HPOI Dataset Designer (HPOIDD) v1.02
User's Guide

Programmer:

Eric C. Sayre
Delta, B.C.
February 24, 2008

If you have any questions about this program, please contact the contract manager:

Marie P. Beaudet
Occupational and Environmental Health Research Studies
Health Statistics Division
Statistics Canada
2200 Main Building Section H
150 Tunney's Pasture Driveway
Ottawa, Ontario K1A 0T6
613-951-7025 (phone)
613-951-0792 (fax)
beaumar@statcan.ca

Table of Contents

Table of Contents 2

1. Introduction 5

1.1 Background 5
1.2 Limitations 6

2. Preparing SAS V9 7

2.1 Setting up SAS for faster execution 7
2.2 Watching the SAS Log window 8
2.3 %Including HPOIDD.SAS 9
2.4 Global macro variables 9
_HPOIDD_LineSize 9
_HPOIDD_PageSize 10
_HPOIDD_MaxPutLines 10
_HPOIDD_ShowEpisodeArgs 10

3. HPOIDD_BigData SAS macro 11

3.1 Background 11
3.2 Macro arguments 11
Inlib 11
Indat 11
Outdat 12
BLZEPoiDup 13
CH1BLPerson 13
3.3 Contents of the output dataset 13

4. HPOIDD_BigData_List_AvailVars SAS macro 16

4.1 Background 16
4.2 HPOI data dictionaries 16

5. HPOIDD_Episode SAS macro 17

5.1 Background 17
5.2 Macro arguments 17
Indat 17
Outdat 17
Outtext 17
SASCodePath 17
EDVName 19
EDVMin 19
EpiOccur 19
DateRange 19
WashTime 20
WashType 20
WashComp 20

SWVName 21
SWVLogic 21
DataDesign 22
Count data *22*
Event-time data *23*
Episode level data *24*
Episode array data *25*
AVarList 26
5.3 THE EPISODES ALGORITHM 29
5.4 SUB-SETTING THE INPUT DATASET FOR TESTING PURPOSES 31

6. Statistical analysis of HPOIDD_Episode and HPOIDD_BigData data 32

6.1 OVERVIEW 32
Caveats of HPOI data 34
6.2 EXAMPLE OF PREPARING THE HPOIDD_BIGDATA DATASET 35
6.3 POISSON REGRESSION 36
Example 1 – GLM fit on Count data in independent hospital-level data records 36
Example 2 – GLM with GEE fit on Count data in repeated measures hospital-level data 39
6.4 LOGISTIC REGRESSION 40
Example 1 – GLM fit on Count data in independent prospective person-level cohort data linked to NPHS 41
Example 2 – GLM with GEE fit on Count data in repeated measures prospective person-level cohort data linked to NPHS 44
Example 3 – GLM fit on per visit HPOIDD_BigData data 45
6.5 LINEAR REGRESSION AND REPEATED MEASURES ANOVA 46
Example 1 – Multiple linear regression fit to summary analysis variable for days of stay in independent EpisodeLevel data 46
Example 2 – Repeated measures ANOVA fit to Count data in linked hospital-level data measured repeatedly over several fiscal years 49
6.6 RETROSPECTIVE CASE-CONTROL DATA 53
Example 1 – Unconditional logistic regression and unstratified odds ratio on Count data in unmatched person-level case-control data (using CCHS for controls) 54
Example 2 – Conditional logistic regression and stratified Mantel-Haenszel odds ratio on Count data in matched person-level case-control data (using CCHS for controls) 57
Example 3 – Person-level case-control Count data matched by a propensity score (using CCHS for controls) 59
6.7 EVENT-TIME MODELS 60
Example 1 – Life table analysis, stratified Kaplan-Meier with log rank test, Cox proportional hazards model, and parametric regressions with exponential and Weibull distributions, on person-level EventTime data linked to CCHS 61
Example 2 – Cox proportional hazards model with time varying covariates on EventTime data in person-level data linked to NPHS 66
Example 3 – Event time modeling from first hospital admission to the next (uses the EpisodeLevel data design) 67
Example 4 – Competing risks event time modeling from the first episode of one type to the first of several competing episodes (uses the EpisodeArray data design) 70

7. References 76

1. Introduction

1.1 Background

The goal of this project was to build a set of user friendly SAS 9.1.3 macros that construct customized analysis-ready datasets from the Hospital Person Orientated Information (HPOI) administrative database, according to a broad spectrum of methodologies. This package of macros is called the HPOI Dataset Designer (HPOIDD).

Before we can consider the data design, we require a basic understanding of the HPOI database from which our data will be constructed. According to the 2005 Statistics Canada article Health Studies Using Linked Administrative Hospital Data[1]:

> There are approximately three million hospital discharges in Canada every year. Each discharge record contains a unique personal linkage ID and includes data on birth date, sex, postal code, hospital, admission and separation dates, diagnoses, procedures and death-in-hospital. This data file is a large potential source of information on disease/procedure rates by person, place and time; health outcomes and hospital utilization.
>
> …
>
> Each hospital collects and codes information on every separation and sends the information to its provinces/territory. All provinces and territories send these files to the Canadian Institute for Health Information (CIHI) every year. They amalgamate similar data from each province/territory into a national Hospital Morbidity file. This file is sent to Statistics Canada.
>
> Statistics Canada uses these records to create and maintain a linkable Health Person-oriented Information (HPOI) hospital Database. The records that create the POI universe are selected by excluding records from newborns and non-residents and records with invalid or blank health numbers. New identification numbers are created to differentiate between parent/child and sex - specific ICD or CCP codes. Values for date of birth/sex/discharge condition are imputed to make them consistent for each health number. In addition health region codes/ ecological census variables are added
>
> Table 1 shows the years and regions available in the Health Person Oriented Information (HPOI) hospital database. From 1994/95 on, linkable data is available for all ten provinces. Quebec is the only province that sends scrambled identification numbers. A change in coding classifications started in 2000/01 and occurred at different times for difference regions. Quebec will not change its coding system until 2006/07.
>
> Table 1. Available hospital data in HPOI Database by Provinces /Territories, Year, Type of Health Number and International Classification of Disease code used (ICD-9, ICD-9-CM, or ICD-10)

	NF	PE	NS	NB	QU	ON	MA	SA	AL	BC	YK	NT	NU
1992/93	9	9		9		9	9						
1993/94	9	9	9	9	9	9	9	9				9	
1994/95	9	9	9	9	9	9	9	9	9	9		9	
1995/96	9	9	9	9	9	9	9	9	9	9		9	
1996/97	9	9	9	9	9	9	9	9	9	9		9	
1997/98	9	9	9	9	9	9	9	9	9	9	9	9	
1998/99	9	9	9	9	9	9	9	9	9	9	9	9	
1999/00	9	9	9	9	9	9	9	9	9	9	9	9	9
2000/01	9	9	9	9	9	9	9	9	9	9	9	9	9
2001/02	10	10	10	9-CM	9	9, 9-CM	9-CM	9, 9-CM, - 10	9-CM	10	10	9-CM	9-CM
2002/03	10	10	10	9-CM	9	10	9-CM	10	10	10	10	10	

: Actual Health Number : Scrambled Health Number

In addition to the HPOI data described above, preliminary work is underway to link these data to Statistics Canada (STC) surveys, e.g., National Population Health Survey (NPHS) and Canadian Community Health Survey (CCHS). HPOIDD macros will not preclude such linkage.

Each separation record in HPOI data corresponds to one visit that ends in separation from the hospital for example through discharge, transfer, or death. A given person may span several records in HPOI over many different years.

The first step in preparing an analysis ready dataset is to define care episodes for each person in the study population (e.g., Canada or a specific province). An "episode" of care can be defined according to the analyst's specifications, and the definition may involve multiple visits within a specified period of time, the application of ICD-9, ICD-9-CM or ICD-10 diagnostic codes, Canadian Classification of Procedures (CCP), Canadian Classification of Interventions (CCI) or user-defined variables constructed out of other HPOI variables. The choice of coding system(s) used will depend on the provinces and years under consideration, and more than one coding system may be required to define an episode type. In that case care must be taken, as there is not a perfect 1 to 1 correspondence between different ICD classification systems.

Exactly how the analysis ready dataset is constructed depends on the data design set up by the analyst in their call to the HPOIDD macros. Available data structures include count data, event time data, single episode data, multiple episode array data, and more. For more details, see the chapter on data designs.

1.2 Limitations

The HPOIDD software package was written to be as general as possible while balancing generality with error checking and user-friendliness. HPOIDD can handle a broad range of analyses. During conceptual development, analysts at Statistics Canada were consulted on what analyses they had performed using HPOI data. All analyses reported during the conceptual development phase can be performed with HPOIDD. However, there will likely be customized analyses in the future that an analyst will want to perform that HPOIDD in its current form cannot do. This program is not everything to everyone. However, a broad range of analytical approaches is covered by the program, and a little flexibility in one's approach should make HPOIDD a powerful time-saving alternative to programming from scratch.

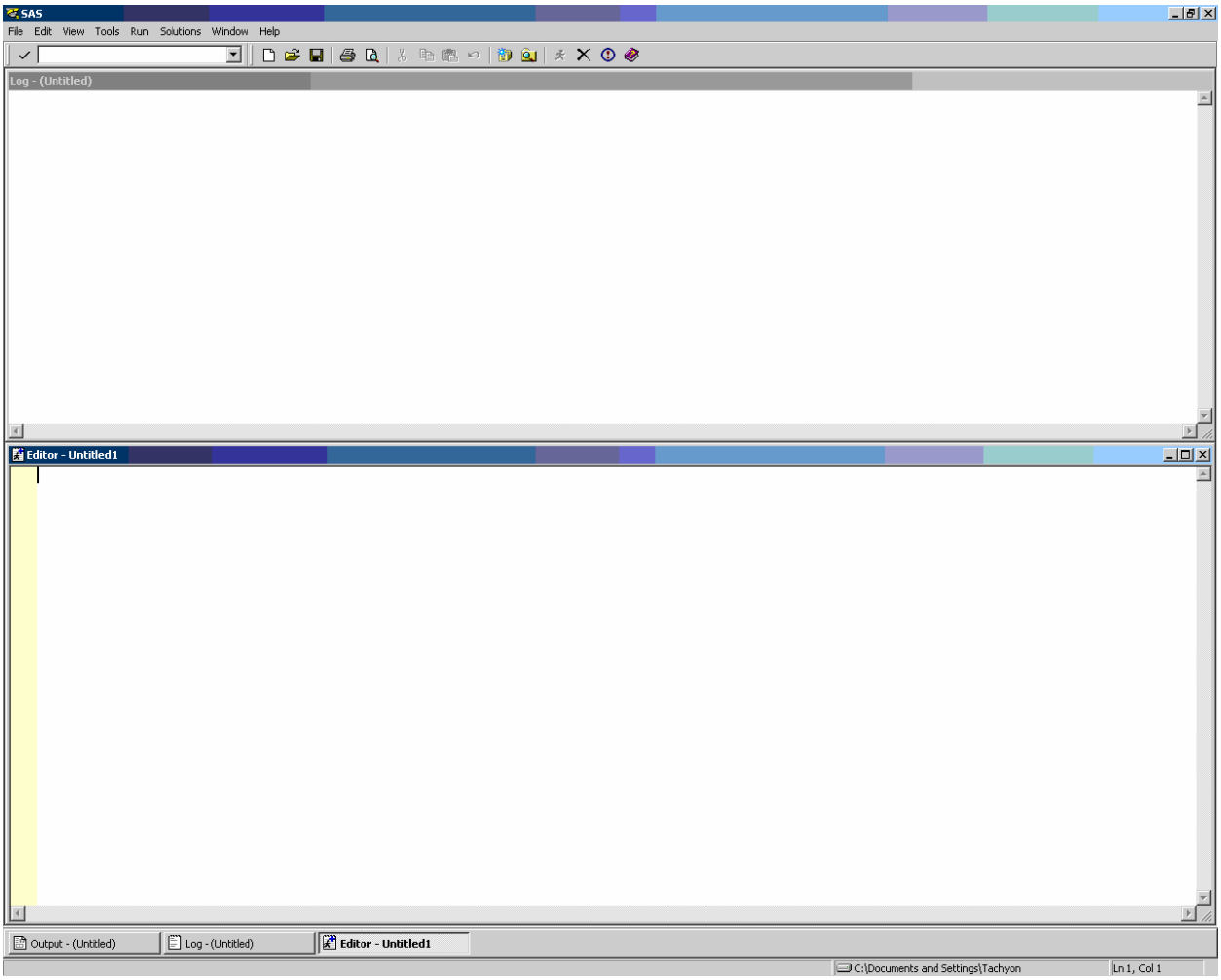
2. Preparing SAS V9

2.1 Setting up SAS for faster execution

To ensure the macros in HPOIDD run as fast as possible, there are a number of things an analyst can do.

Firstly, whenever possible, store large SAS datasets on local hard drives rather than network drives. Depending on the speed of the network and traffic during the day, this can speed up runs substantially. Also, it can help to avoid conflicts between multiple analysts trying to read the same large SAS dataset at the same time (which would lead to an error message for all but one of the analysts).

Next, ensure that the SAS Explorer and Results windows are closed during execution. To ensure that they are not merely minimized, check the SAS task bar at the bottom of the SAS GUI (graphical user interface). Note that in following image, the SAS task bar only displays the Program Editor, Log and Output windows, not the Explorer or Results windows. The Explorer and Results windows open with SAS by default unless you change a setting in the sasv9.cfg file from "-dmsexp" to "-nodmsexp". Therefore unless your network administrator makes that change to the sasv9.cfg file, you will have to manually close the Explorer and Results windows before running the HPOIDD macros. Failure to close the Explorer and Results windows may result in runs taking several times longer.

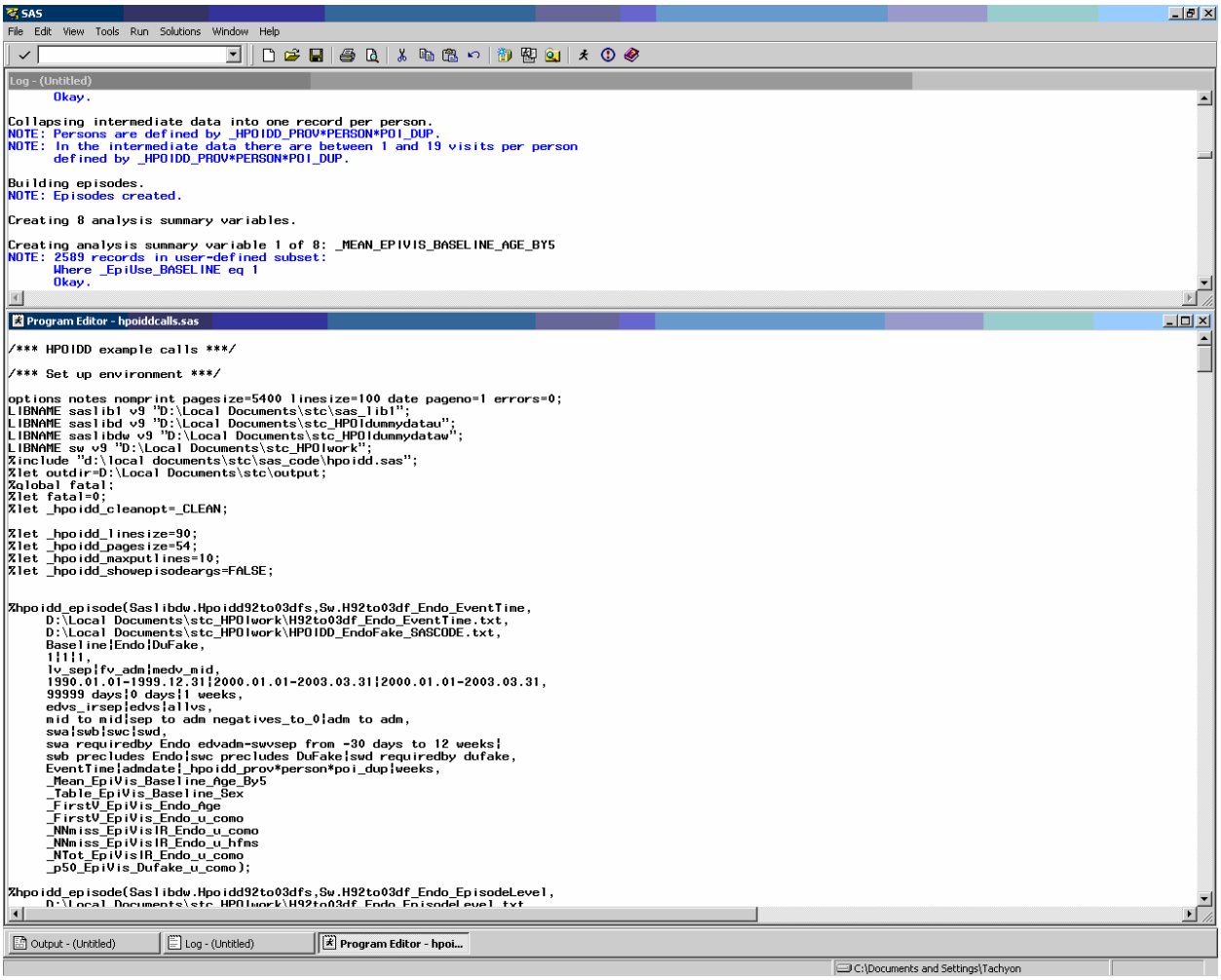


Thirdly, it is sometimes helpful to reload SAS fresh after many runs, as SAS can slow down over time for various reasons. Within a full day of intensive runs, SAS may slow down by a factor in the tens or more. Reload SAS every couple hours and this should not be an issue.

Finally, for fastest execution, screensavers should be disabled and SAS should be running in the foreground. Failure to disable screensavers and/or have SAS running in the foreground could result in runs taking several times longer.

2.2 Watching the SAS Log window

The macros in HPOIDD provide detailed feedback in real time in the SAS Log window. This includes notes, warnings and error messages. It is highly recommended to keep the SAS Log window visible while running any HPOIDD macro. The following screen shows an example of such feedback.



2.3 %Including HPOIDD.SAS

Before the macros in HPOIDD.SAS can be called, the program must be %included in SAS. Example:

```
%include "d:\local documents\stc\sas_code\hpoidd.sas";
```

2.4 Global macro variables

After %including HPOIDD.SAS, there are 4 global macro variables that can be changed to alter the action of the program. Every time the program is %included, these variables are reset to their defaults, so you will need to change them again if you are using custom values. These variables are as follows.

_HPOIDD_LineSize

This is the linesize setting in SAS. It should be an integer between 90 and 256. Default is 90. Example:

```
%include "d:\local documents\stc\sas_code\hpoidd.sas";
%let _HPOIDD_LineSize=120;
```

_HPOIDD_PageSize

This is the pagesize setting in SAS. It should be an integer between 20 and 32767. Default is 54.
Example:

```
%include "d:\local documents\stc\sas_code\hpoidd.sas";
%let _HPOIDD_PageSize=10000;
```

_HPOIDD_MaxPutLines

This is the maximum number of lines to print in the SAS Log window for any given note, warning or error message. This should be an integer between 10 and 32767. Default is 32767. Many HPOIDD messages are comprised of several individual smaller message portions. Each individual smaller message portion will be truncated according to _HPOIDD_MaxPutLines; all individual smaller message portions comprising the larger message will be shown at least in part. Where the current value of _HPOIDD_MaxPutLines is &_HPOIDD_MaxPutLines, if any given individual smaller message portion is truncated, that message portion will be followed by "... (message truncated per _HPOIDD_MaxPutLines=&_HPOIDD_MaxPutLines)".
Example:

```
%include "d:\local documents\stc\sas_code\hpoidd.sas";
%let _HPOIDD_MaxPutLines=10;
```

_HPOIDD_ShowEpisodeArgs

This should equal TRUE or FALSE to indicate whether or not errors encountered in the HPOIDD_Episode macro should result in the printing in the Log window the explanation of the episode algorithm found section *5.3 The episodes algorithm*. Default is FALSE.
Example:

```
%include "d:\local documents\stc\sas_code\hpoidd.sas";
%let _HPOIDD_ShowEpisodeArgs=True;
```

3. HPOIDD_BigData SAS macro

3.1 Background

The first step is to run this macro, which creates a single flat file dataset from a library with separate HPOI data sets from all available fiscal years. To call the HPOIDD_BigData macro, submit a statement similar to this one, where the macro arguments are customized according to the explanations in the next section.

```
%HPOIDD_BigData(inlib,indat,outdat,blzepoidup,ch1blperson);
```

3.2 Macro arguments

Inlib

INLIB is the input SAS library containing the HPOI SAS datasets. This library should contain all the HPOI SAS files.

Indat

INDAT is a list of HPOI SAS datasets to use. To utilize all available HPOI data in the INLIB library, set INDAT argument to _ALLDATA.
Notes:
i) Input SAS dataset names cannot begin with the _ symbol if they are in the WORK library.
ii) All input datasets must start with the prefix CAN, DIAGNOSIS or INTERVENTION. The next four characters in the dataset name should indicate the fiscal year. For example, CAN9394 should contain CAN data from fiscal year 1993/1994. There can be any additional characters appended to the dataset name, for example the name DIAGNOSIS0304DF might be used for a dummy DIAGNOSIS dataset for fiscal year 2003/2004.
iii) At least one CAN dataset is required.

iv) All CAN input datasets must contain the currently recognized variables for these data, which include: DATA_YR, PROV and SEP_NUM.
v) All DIAGNOSIS input datasets must contain the currently recognized variables for these data, which include: SEP_NUM and DIAG_SEQ_ID.
vi) All INTERVENTION input datasets must contain the currently recognized variables for these data, which include: SEP_NUM, EPISODE_SEQ_ID and INTERVENTION_SEQ_ID.

vii) Records in each CAN dataset prior to 199596 should be uniquely identified by PROV*SEP_NUM.
viii) Records in each CAN dataset from 199596 and later should be uniquely identified by SEP_NUM.
ix) Records in each DIAGNOSIS dataset should be uniquely identified by SEP_NUM*DIAG_SEQ_ID.
x) Records in each INTERVENTION dataset should be uniquely identified by SEP_NUM*EPISODE_SEQ_ID*INTERVENTION_SEQ_ID.

xi) All records in CANyyyy (where yyyy is the four-digit fiscal year) should contain the variable DATA_YR set identically to the six-digit same fiscal year representation. For example, CAN9293 should contain DATA_YR set identically to 199293, while CAN0304 should contain DATA_YR set identically to 200304. Even though no data are currently expected before 199293, in case earlier data should arise in the future, 4-digit fiscal years starting with an 8 will be treated as occurring in the 1980s.

xii) The HPOIDD_BigData macro will create a DATA_YR variable on the combined INTERVENTION datasets matching the fiscal year indicated in each dataset name. Every combination of DATA_YR*SEP_NUM in the combined INTERVENTION datasets must also be found in the combined CAN datasets.
xiii) The HPOIDD_BigData macro will create a DATA_YR variable on the combined DIAGNOSIS datasets matching the fiscal year indicated in each dataset name. Every combination of DATA_YR*SEP_NUM in the combined DIAGNOSIS datasets must also be found in the combined CAN datasets.

xiv) Although the above mentioned variables are recorded as numeric data type on some datasets and character type on others, all are expected to contain numbers, and as such will be converted to numbers for merging purposes.

xv) SEP_NUM sequences may be regenerated each data year. Therefore, in combined data output by this macro, records will be uniquely identified by DATA_YR*SEP_NUM.

xvi) PERSONs are identified by the combinations of _HPOIDD_Prov*Person*POI_Dup where _HPOIDD_Prov is Prov converted to numeric type. Pre-200102, this is sufficient. In 200102 and later CAN datasets, HEALTH_CARD_PROV_CODE is checked against Prov according to the following map:

```
PROV HEALTH_CARD_PROV_CODE (specific years)
10   NL                    200304 and later
10   NF                    200102 and 200203
11   PE
12   NS
13   NB
24   QC
35   ON
46   MB
47   SK
48   AB
59   BC
60   YT
61   NT
62   NU
```

Any separation records that fail this check are dropped according to the document Data_Dictionary_CANxxxx&Abstract_v2004.doc.

Outdat

OUTDAT is the name (with SAS library) of the output dataset. Each record will contain all the available information on one separation. Records will be uniquely identified by _HPOIDD_DATA_YR*_HPOIDD_SEP_NUM.

Note:
- Output SAS dataset names cannot begin with the _ symbol if they are in the WORK library.

BLZEPoiDup

Set BLZEPOIDUP to CHANGE_POI_DUP_BLANKS_TO_ZEROS if you want to set blank POI_Dup values to 0. Set BLZEPOIDUP to NO_CHANGE_POI_DUP_BLANKS to leave blank POI_Dup as is. In the latter case, those records will be dropped, as persons are identified by _HPOIDD_Prov*Person*POI_Dup where _HPOIDD_Prov is Prov converted to numeric type.

CH1BLPerson

Set CH1BLPERSON to CHANGE_PERSON_CH1_TO_BLANK if you want to set PERSON values of a single character to blank. Set CH1BLPERSON to NO_CHANGE_PERSON_CH1_TO_BLANK to leave PERSON values of a single character as is. In the latter case, those records will be kept and the single character will be treated as a valid PERSON identifier, as persons are identified by _HPOIDD_Prov*Person*POI_Dup where _HPOIDD_PROV is PROV converted to numeric type.
WARNING:
Setting CH1BLPERSON to NO_CHANGE_PERSON_CH1_TO_BLANK can result in a large number of apparent separations for each "person" with PERSON value a single character (e.g., "0"). It is probable that PERSON values of a single character are actually unidentified and as such it is recommended to set CH1BLPERSON to CHANGE_PERSON_CH1_TO_BLANK.

3.3 Contents of the output dataset

The following variables are available in the output dataset created by HPOIDD_BigData:

i) All variables from the DIAGNOSIS datasets are available under the same name but as arrays, so with numeric suffixes inside curly brackets {} appended, ranging from 1 to _HPOIDD_DIAGNOSIS_ALEN, where _HPOIDD_DIAGNOSIS_ALEN is a variable containing the number of diagnoses for the current record (DATA_YR*SEP_NUM) in the HPOIDD_BigData dataset. Arrays will be set up automatically, so do not include array statements for these variables in your argument, and do not refer to array elements beyond _HPOIDD_DIAGNOSIS_ALEN. Available DIAGNOSIS variables include _HPOIDD_DIAGNOSIS_ALEN (numeric), and where i ranges from 1 to _HPOIDD_DIAGNOSIS_ALEN:

```
Variable               Type Length Note
_HPOIDD_DIAG_SEQ_ID{i} Num  8      Converted to numeric by DIAG_SEQ_ID+0.
DIAG_CM_CODE{i}        Char 5
DIAG_ICD10_CODE{i}     Char 7
DIAG_ICD9_CODE{i}      Char 6
DIAG_PREFIX{i}         Char 1
DIAG_TYPE_CODE{i}      Char 1
```

ii) All variables from the INTERVENTION datasets are available under the same name but as arrays, so with numeric suffixes inside curly brackets {} appended, ranging from 1 to _HPOIDD_INTERVENTION_ALEN, where _HPOIDD_INTERVENTION_ALEN is a variable containing the number of diagnoses for the current record (DATA_YR*SEP_NUM) in the HPOIDD_BigData dataset. Arrays will be set up automatically, so do not include array

statements for these variables in your argument, and do not refer to array elements beyond _HPOIDD_INTERVENTION_ALEN. Available INTERVENTION variables include _HPOIDD_INTERVENTION_ALEN (numeric), and where i ranges from 1 to _HPOIDD_INTERVENTION_ALEN:

```
Variable                        Type Length Note
_HPOIDD_EPISODE_SEQ_ID{i}       Num  8      Conv. to Num by EPISODE_SEQ_ID+0.
_HPOIDD_INTERVENTION_SEQ_ID{i} Num  8      Conv. to Num by
                                           INTERVENTION_SEQ_ID+0.
EXTENT_ATTRIBUTE{i}             Char 2
INTERVENTION_CCI_CODE{i}        Char 10
INTERVENTION_CCP_CODE{i}        Char 4
INTERVENTION_CM_CODE{i}         Char 4
INTERVENTION_SUFFIX{i}          Char 1
LOCATION_ATTRIBUTE{i}           Char 2
STATUS_ATTRIBUTE{i}             Char 2
```

iii) All variables specific to the pre-200102 CAN datasets are available under the same name only if there were pre-200102 CAN datasets in the data comprising the HPOIDD_BigData dataset:

```
Variable Type Length
ACC_1    Char 7
ACC_2    Char 7
ACC_3    Char 7
ACC_4    Char 7
ACC_5    Char 7
ACC_LOC  Char 1
DIS_OLD  Char 1
DISCHARG Char 1
EAREAS   Char 8
LINKVAR  Char 21
NEWBORN  Char 1
SEX_FLAG Char 1
```

iv) All variables common to pre-200102 and 200102 or later CAN datasets are available under the same name:

```
Variable     Type Length Note
PERSON       Char 12     Leading spaces are removed by the macro.
ACUTE        Char 1
ADMDATE      Num  8
AGE          Num  8
AGE_BY5      Char 3
AGE_CODE     Char 1
AGE_DIAG     Char 2
AGE_SURG     Char 3
BTHDATE      Num  8
BTHDATE_OLD  Num  8
CDL_CODE     Char 3
CH_FLAG      Char 1
CHP_DIAG     Char 2
CHP_SURG     Char 3
CODING_CLASS Char 1
CPL_CODE     Char 3
DAYS_ST      Num  8
EXCLUS       Char 1
```

```
HOSP_NO      Char 5
ICFMI_NO     Char 5
ID_OLD       Char 12
IMPUTED      Char 3
OOP_FLAG     Char 1
POI_DUP      Char 1
POSTAL       Char 6
PRIMSERV     Char 2
RES_FLAG     Char 1
RESPON       Char 2
SEPDATE      Num  8
SEX          Char 1
SEX_OLD      Char 1
SGC          Char 7
VISIT        Char 4
```

v) All variables specific to the 200102 or later CAN datasets are available under the same name only if there were 200102 or later CAN datasets in the data comprising the HPOIDD_BigData dataset:

```
Variable                    Type Length
ADMISSION_CATEGORY          Char 1
DAUID                       Char 8
DISCHARGE_DISP_POI          Char 2
DISCHARGE_DISPOSITION       Char 2
ENTRY_CODE                  Char 1
ERR_FLAG                    Char 1
HEALTH_CARD_PROV_CODE       Char 2
HOSPITAL_TYPE               Char 1
MR_DIAG_CM_CODE             Char 5
MR_DIAG_ICD10_CODE          Char 7
MR_DIAG_ICD9_CODE           Char 5
PRINC_INTERVENTION_CCI_CODE Char 10
PRINC_INTERVENTION_CCP_CODE Char 4
PRINC_INTERVENTION_CM_CODE  Char 4
PRINC_INTERVENTION_SUFFIX   Char 1
```

vi) The following additional special HPOIDD variables are also available:

```
Variable        Type Length Note
_HPOIDD_DATA_YR Num  8      This is conv. to Num by DATA_YR+0.
_HPOIDD_PROV    Num  8      This is conv. to Num by PROV+0.
_HPOIDD_SEP_NUM Num  8      This is conv. to Num by SEP_NUM+0
                            in 199596 or later data and by
                            SEP_NUM+PROV*100000000 in pre-199596.
_HPOIDD_INFO    Char 256    The HPOIDD version number used to create the
                            dataset, as well as additional information
                            about the macro call and the data.
```

4. HPOIDD_BigData_List_AvailVars SAS macro

4.1 Background

This macro lists the available variables that the user may reference in their SAS code arguments in other HPOIDD macros. It simply prints in the log window the information presented in the HPOIDD_BigData chapter above, subsection *Contents of the output dataset*. To call the HPOIDD_BigData_List_AvailVars macro, submit the following statement. (There are no macro arguments to customize.)

```
%HPOIDD_BigData_List_AvailVars;
```

4.2 HPOI data dictionaries

There are four data dictionaries that should accompany the HPOIDD program[2,3,4,5]. The latest versions should always be consulted. At the time this program is being developed, the latest available data dictionaries are

- Person Oriented Information and Hospital Morbidity Data Dictionary. Health Statistics Division, Statistics Canada. Prepared April, 1999, Updated March 27, 2003. File name: Hospital POI Data Dictionary.doc[2]
- Combined HPOI & HMDB Data Dictionary Data years: Fiscal 2001 to Fiscal 2004. Health Statistics Division, Statistics Canada. File name: Data_Dictionary_CANxxxx&Abstract_v2004.doc[3]
- Hospital Morbidity Database (HMDB) From Fiscal 2001 HMDB Data Dictionary for: Diagnosis Table. Health Statistics Division, Statistics Canada. File name: Data_Dictionary_Diagnosis_v2004.doc[4]
- Hospital Morbidity Database (HMDB) From Fiscal 2001 HMDB Data Dictionary for: Intervention Table. File name: Data_Dictionary_Intervention_v2004.doc[5]

Respectively, these contain explanations of the HPOI variables in pre-200001 CAN flat file datasets, 200102 or later CAN relational datasets, 200102 or later Diagnosis relational datasets, and 200102 or later Intervention relational datasets.

5. HPOIDD_Episode SAS macro

5.1 Background

The next step is to run this macro, which creates the analysis dataset from the combined HPOI dataset produced by HPOIDD_BigData. This macro produces for each _HPOIDD_Prov*Person*POI_Dup combination an array of episodes, then organizes the data according to the user specified data design. The output dataset can be analyzed by an appropriate SAS procedure. Specific details of the supported data structures are given in the subsections on each macro argument, below. To call the HPOIDD_Episode macro, submit a statement similar to this one, where the macro arguments are customized according to the explanations in the next section.

```
%HPOIDD_Episode(indat,outdat,outtext,sascodepath,
                edvname,edvmin,epioccur,
                daterange,washtime,washtype,washcomp,
                swvname,swvlogic,datadesign,avarlist);
```

5.2 Macro arguments

Indat

INDAT is the name (with SAS library) of the input SAS dataset that was produced by the HPOIDD_BigData macro.
Note:
- Input SAS dataset names cannot begin with the _ symbol if they are in the WORK library.

Outdat

OUTDAT is the name (with SAS library) of the output SAS dataset that will be ready for analysis.
Note:
- Output SAS dataset names cannot begin with the _ symbol if they are in the WORK library.

Outtext

OUTTEXT is the full path of an output text file which will be written containing information about the output dataset.

SASCodePath

SASCODEPATH is the path to the text file containing the analyst-prepared SAS code. The SAS code in this file, executed directly on the input HPOIDD_BigData dataset, can define a subset of interest, should define the 0-1 episode-defining visit (EDV) variables named in EDVNAME, and should define all the special washout visit (SWV) variables referred to in the macro call, if any. After running this SAS code, the EDV variables and all SWV variables must equal 0 or 1 for each separation in the input HPOIDD_BigData dataset. Do not include data statements, array definitions for available HPOIDD_BigData variables, or run statements, as these will be defined already in the data step. For a list of the available variables in the HPOIDD_BigData datasets

that you may refer to in your code, refer to the HPOIDD user's guide or submit the following statement in SAS:

```
%HPOIDD_BigData_List_AvailVars;
```

The data step into which the SAS code will be inserted is:

```
data _use;
 set &inlib..&indat;
 by _HPOIDD_PROV PERSON POI_DUP ADMDATE SEPDATE
    _HPOIDD_DATA_YR _HPOIDD_SEP_NUM;
 array _HPOIDD_DIAG_SEQ_ID{&max_HPOIDD_DIAGNOSIS_ALEN};
 array DIAG_CM_CODE{&max_HPOIDD_DIAGNOSIS_ALEN} $;
 array DIAG_ICD10_CODE{&max_HPOIDD_DIAGNOSIS_ALEN} $;
 array DIAG_ICD9_CODE{&max_HPOIDD_DIAGNOSIS_ALEN} $;
 array DIAG_PREFIX{&max_HPOIDD_DIAGNOSIS_ALEN} $;
 array DIAG_TYPE_CODE{&max_HPOIDD_DIAGNOSIS_ALEN} $;
 array _HPOIDD_EPISODE_SEQ_ID{&max_HPOIDD_INTERVENTION_ALEN};
 array _HPOIDD_INTERVENTION_SEQ_ID{&max_HPOIDD_INTERVENTION_ALEN};
 array EXTENT_ATTRIBUTE{&max_HPOIDD_INTERVENTION_ALEN} $;
 array INTERVENTION_CCI_CODE{&max_HPOIDD_INTERVENTION_ALEN} $;
 array INTERVENTION_CCP_CODE{&max_HPOIDD_INTERVENTION_ALEN} $;
 array INTERVENTION_CM_CODE{&max_HPOIDD_INTERVENTION_ALEN} $;
 array INTERVENTION_SUFFIX{&max_HPOIDD_INTERVENTION_ALEN} $;
 array LOCATION_ATTRIBUTE{&max_HPOIDD_INTERVENTION_ALEN} $;
 array STATUS_ATTRIBUTE{&max_HPOIDD_INTERVENTION_ALEN} $;
 ******************************;
 * User-supplied SAS code follows;
 %include "&sascodepath";
 run;
```

The following lines are also run to check that all variables listed on EDVNAME and SWVNAME arguments are non-missing and evaluate to 0 or 1 after the above data step:

```
data _hpoidd_bad_edv_or_swv;
 set _use;
 _hpoidd_bad_edv_or_swv=1;
 if 0 eq 1 then output;
 %do i=1 %to &numedv;
  else if &&edvname&i ~in (0 1) then output;
 %end;
 %do i=1 %to &numswv;
  else if &&swvname&i ~in (0 1) then output;
 %end;
 run;
```

Notes:

i) The following HPOIDD_BigData variables are restricted and must not be altered during execution of your SAS code:

_HPOIDD_PROV PERSON POI_DUP ADMDATE SEPDATE _HPOIDD_DATA_YR _HPOIDD_SEP_NUM

The following additional variable names are restricted and much not be referenced:

_BAK__HPOIDD_PROV _BAK_PERSON _BAK_POI_DUP _BAK_ADMDATE _BAK_SEPDATE _BAK__HPOIDD_DATA_YR _BAK__HPOIDD_SEP_NUM _MIDDATE _PERSONI _ORDER _NUMVISITS _MAX_HPOIDD_BAD_EDV_OR_SWV _HPOIDD_BAD_EDV_OR_SWV

ii) You must not use macro code in your SAS code.
iii) The by statement will allow retain statements and the special SAS first. and last. variables to be used to keep track of previous visit results on a per person basis as the data step runs. For an example, see section *6.5 LINEAR REGRESSION AND REPEATED MEASURES ANOVA Example 1 – Multiple linear regression fit to summary analysis variable for days of stay in independent EpisodeLevel data*.

EDVName

EDVNAME is a | delimited list of the variable names of the episodes. These names cannot contain an _ symbol or end with an integer, and cannot exceed 16 characters in length. There can be many but must be at least one episode name.
Example:

```
%HPOIDD_Episode(...,MyoInfarct|Pacemaker|Death,...);
```

EDVMin

EDVMIN is a | delimited list of integers >=1. The ith integer defines the minimum number of the ith type EDV that must occur in an episode for the episode to be counted. The most common entry in EDVMIN might be 1, indicating that a single EDV in an episode of visits is enough to define that episode as an ith type episode. There must be one integer for each EDV listed in EDVNAME.
Example:

```
%HPOIDD_Episode(...,1|1|1,...);
```

EpiOccur

EPIOCCUR should be set to a | delimited list of keywords, one keyword for each episode named in EDVNAME. Keywords should be one of FV_ADM, FV_SEP, FV_MID, LV_ADM, LV_SEP, LV_MID, MV_ADM, MV_SEP, MV_MID, FEDV_ADM, FEDV_SEP, FEDV_MID, LEDV_ADM, LEDV_SEP, LEDV_MID, MEDV_ADM, MEDV_SEP or MEDV_MID, to indicate on what date an episode of visits should be deemed to occur. Prefix FV indicates first visit in the episode regardless of the visit's EDV status, LV indicates last visit in the episode regardless of the visit's EDV status, MV indicates the middle visit in the episode regardless of the visit's EDV status, FEDV indicates first EDV in the episode, LEDV indicates last EDV in the episode, and MEDV indicates the middle EDV in the episode. In the case of an even number visits considered, "middle visit" or "middle EDV" is taken to mean the first of the two middle visits or EDVs. Suffix ADM indicates admission date, SEP indicates separation date, and MID indicates midpoint date between admission and separation (rounded down in the case of half-days).
Example:

```
%HPOIDD_Episode(...,LV_SEP|FEDV_MID|LEDV_SEP,...);
```

DateRange

DATERANGE is a | delimited list of date ranges, one date range for each episode named in EDVNAME. The ith date range in the list should indicate the range of dates in which the ith episode must occur to be counted. The format of each date range is YYYY.MM.DD-YYYY.MM.DD, where YYYY indicate a 4-digit year, MM indicates a 2-digit month, and DD indicates a 2-digit day. To set no lower bound on a date range, set the lower date in the range to

1900.01.01. This is the earliest allowable admission or separation date in HPOIDD. To set no upper bound on a date range, set the upper date in the range to 2075.01.01. This is the latest allowable admission or separation date in HPOIDD. Valid dates are between 1900.01.01 and 2075.01.01.
Example:

```
%HPOIDD_Episode(...,1996.11.01-2000.03.31|1900.01.01-1996.11.01|1995.01.01-
2075.01.01,...);
```

WashTime

WASHTIME is a | delimited list of washout times, one washout time for each episode named in EDVNAME. Each time should be an integer and then after a space the unit, either days or weeks. The washout time is the minimum time that must pass in order for a subsequent visit to be counted as a new health encounter rather than an extension of the previous visit.
Warning:
Due to the possibility of partially or fully overlapping visits (in real data, this happens), negative numbers are possible depending on how the comparison between visits is being made according to the WASHCOMP argument. This is because the admission, midpoint or separation date of the "next" visit may actually fall before the separation date of the current visit. If you set a given WASHTIME to 0 days but specify on WASHCOMP to make comparisons from current separation date to the next visit's admission date for example, then despite the washout time of 0 days two adjacent visits may still be combined into one visit when building an episode if the time comparison is negative. For this reason, it is important to use the NEGATIVES_TO_0 option on the WASHCOMP argument when you want all visits to be treated as distinct care episodes. In that case, negative time comparisons will be treated as 0.
Example:

```
%HPOIDD_Episode(...,-5 days|0 days|52 weeks,...);
```

WashType

WASHTYPE is a | delimited list of keywords, one keyword for each episode named in EDVNAME. Set the ith keyword to ALLVS if in the ith EDV type episode all visits (whether or not they satisfy the EDV, any optional special washout visits affecting that EDV, and whether or not the visit itself is in range) should potentially contribute to the ith EDV type episode, or set the ith keyword to EDVS if only EDVs that pass the special washout checks (whether or not the visit itself is in range) should contribute to the ith EDV type episode. Set the ith keyword to ALLVS_IRSEP or EDVS_IRSEP to utilize the above definitions with the difference that only in-range visits (according to SEPDATE) should be counted towards the ith type episode. Set the ith keyword to ALLVS_IRADM or EDVS_IRADM to utilize the above definitions with the difference that only in-range visits (according to ADMDATE) should be counted towards the ith type episode.
Example:

```
%HPOIDD_Episode(...,ALLVS|ALLVS|EDVS,...);
```

WashComp

WASHCOMP is a | delimited list of 3- or 4-word sentences, one sentence for each episode named in EDVNAME. These define how the washout time for that EDV is to be calculated. In every sentence, the second word should be TO. The keyword before TO indicates from what date during the current visit to make the time comparison, using keyword ADM for admission date,

SEP for separation date, and MID for the midpoint date between admission and separation (rounded down in the case of half-days). The keyword after TO indicates on what date during the subsequent visit the comparison should be made. For example, the first three words being SEP TO ADM means the washout time is compared to the time difference between the current visit's separation date and the subsequent visit's admission date. Optionally, the word NEGATIVES_TO_0 can be added onto the end of the sentence. If so, negative comparisons (as can happen with overlapping visits) will be treated as 0 days. This can be useful if you want to ensure that every visit is treated as distinct (e.g., if you set the corresponding WASHTIME argument to 0 days).
Warning:
Due to the possibility of partially or fully overlapping visits, negative numbers are possible depending on how the comparison between visits is being made according to the WASHCOMP argument. This is because the admission, midpoint or separation date of the "next" visit may actually fall before the separation date of the current visit. If you set a given WASHTIME to 0 days but specify on WASHCOMP to make comparisons from current separation date to the next visit's admission date for example, then despite the washout time of 0 days two adjacent visits may still be combined into one visit when building an episode if the time comparison is negative. For this reason, it is important to use the NEGATIVES_TO_0 option on the WASHCOMP argument when you want all visits to be treated as distinct care episodes. In that case, negative time comparisons will be treated as 0.
Example:

```
%HPOIDD_Episode(...,SEP to ADM|SEP to SEP|adm to mid NEGATIVES_TO_0,...);
```

SWVName

SWVNAME is a | delimited list of the variable names of special washout indicators. These names cannot contain an _ symbol or end with an integer, and cannot exceed 16 characters in length. There may be many special washout variables. If there are no special washout variables, set this to _NOSWV.
Example:

```
%HPOIDD_Episode(...,SpecWashA|SpecWashB,...);
```

SWVLogic

SWVLOGIC is a | delimited list of sentences defining how each episode variable is to be affected by each special washout variable. There should be one sentence for each relationship defined. Each SWV must affect at least one EDV and may affect many EDVs. Each sentence is made up of either 3 or 10 words. The first word is the SWV variable name. The second word is a keyword, either PRECLUDES or REQUIREDBY. The third word is the EDV in the relationship. If the chronological order and time difference between the SWV and EDV doesn't matter, then the sentence can end there. Otherwise, 7 more words must be added to the sentence: EDVSWVTIMEDIFF from STATIME STAUNITS to ENDTIME ENDUNITS
EDVSWVTIMEDIFF defines which visit dates are to be compared in this relationship. Valid keyword pairs for EDVSWVTIMEDIFF are EDVADM-SWVADM, EDVADM-SWVSEP, EDVADM-SWVMID, EDVSEP-SWVADM, EDVSEP-SWVSEP, EDVSEP-SWVMID, EDVMID-SWVADM, EDVMID-SWVSEP and EDVMID-SWVMID. The first three letters of each keyword indicate that the difference is calculated as EDV visit date minus SWV visit date. The last thee letters of each keyword indicate whether the EDV and SWV visit dates are to be

taken as (ADM) admission date, (SEP) separation date, or (MID) midpoint date between admission and separation (rounded down in the case of half-days). STATIME and ENDTIME are integers, positive or negative, and STAUNITS and ENDUNITS are units, either days or weeks. This describes the range in which the EDV must occur relative to the SWV to satisfy that relationship. As implied by the form of the sentence, if the EDV occurrence date minus the SWV occurrence date as indicated by the keywords on this argument falls inside the indicated range, then the special washout relationship is satisfied. If there are no special washout variables, set this to _NOSWV.
Example:

```
%HPOIDD_Episode(...,
   SpecWashA precludes MyoInfarct EDVADM-SWVSEP from -30 days to 12 weeks|
   SpecWashB requiredby MyoInfarct|
   SpecWashB requiredby Pacemaker EDVADM-SWVMID from 0 days to 15 days,...);
```

DataDesign

DATADESIGN specifies the data design. Supported data designs include count data, event-time data, episode level data and episode array data. In count data, episodes are counted within each experimental unit. In event time data, the time to the occurrence of the first episode is recorded. In either case output HPOIDD_BigData data must be specified to have an experimental unit identified by hospital, province, health region, person (identified by _HPOIDD_Prov*Person*POI_Dup where _HPOIDD_Prov is Prov converted to numeric type), or any other variable combination superseding (coarser than) visits in the data. In episode level data, each output record contains an episode and there may be none, one or multiple episodes per person. In episode array data, each output record is for a single person, and many different episode definitions contribute to one large overall array of episodes, ordered by episode date.

The appropriate design depends on the planned analysis. This argument is a | delimited list of sub-arguments, as follows.

Count data

i) First is a keyword to designate the type of data design. Set this to COUNT for Count data, which contains counts of episodes.
ii) Next (after a delimiting | symbol) is a keyword to indicate when a visit should be deemed to occur for purposes of creating summary analysis variables and for determining if a visit is in-range. Set this to either ADMDATE or SEPDATE. Note that when the <episodes> occur is specified separately for each EDV episode type on the EPIOCCUR argument.
iii) Next (after a delimiting | symbol) is a combination of HPOI variables (delimited by the * symbol) to uniquely define the experimental unit in which to group and count episodes. For example, _HPOIDD_Prov*Person*POI_Dup uniquely identifies persons and _HPOIDD_Prov*HOSP_No uniquely identifies hospitals.
_HPOIDD_DATA_YR*_HPOIDD_SEP_NUM uniquely identifies HPOI separation records, where _HPOIDD_SEP_NUM contains additional information about province for those data years when SEP_NUM was not unique across provinces.
iv) Next (after a delimiting | symbol) is a keyword to indicate the time unit in which to group and count episodes. This should be set to either TotalTime, CalendarYear, FiscalYear or Month. TotalTime indicates that the counts should be tabulated over the valid date range portion of all available years of data; there will be one record per experimental unit in this case. Otherwise, for

each experimental unit there may be several records, one record per calendar year, fiscal year or month.

The output dataset will contain a count of each episode type for each experimental unit-time unit combination. The following special variables will be added to the output dataset where EDVNAME is each episode variable name defined earlier. _Count_EDVNAME, _PersAtRisk_EDVNAME and _PTAtRisk_EDVNAME are created to contain the count, total number of persons at risk and total person-time at risk (in days) of the episode named EDVNAME for the experimental unit-time unit combination of that record. The latter variable will be important for calculating incidence rates, or for performing generalized linear modeling such as Poisson regression. _RecStaDate_EDVNAME and _RecEndDate_EDVNAME will contain the first and last dates of consideration for that output record's time unit, which is based in part on the valid episode date range, and in part on the time unit (e.g., fiscal year). _PersAtRisk_EDVNAME is the count of all persons with a visit occurring between _RecStaDate_EDVNAME and _RecEndDate_EDVNAME. When a visit occurs is based on the DATADESIGN specification. _PTAtRisk_EDVNAME for an experimental unit will be simply calculated as _PTAtRisk_EDVNAME=_PersAtRisk_EDVNAME*(_RecEndDate_EDVNAME-_RecStaDate_EDVNAME)

If the time unit was specified as CalendarYear, a variable _CalendarYear is created to contain the 4-digit calendar year for each output HPOIDD record. If the time unit was specified as FiscalYear, a variable _FiscalYear is created to contain the 6-digit fiscal year for each output HPOIDD record (for example, 1999/2000 fiscal year is recorded as 199900). If the time unit was specified as Month, a variable _YearMonth is created to contain the 6-digit year-month for each output HPOIDD record (for example, March 1999 is recorded as 199903).

Application:
Count data can be used in various analyses including incidence rates, odds ratios, Poisson regression or other generalized linear models (GLMs) such as binary or ordinal logistic regression, repeated measures ANOVA if counts are high enough for a normal approximation, generalized estimating equations (GEE) versions of the aforementioned GLMs—for correlated data when there are multiple records on a given experimental unit—and more.

Example:
```
%HPOIDD_Episode(...,Count|AdmDate|Person|TotalTime,...);
```

Event-time data

Event-time data are also commonly known as "lifetime" or "survival time" data, however we will avoid those terms here since in conjunction with health-related data like HPOI data they could lead to confusion. Event-time data refers to the time to an event, and in the context of HPOIDD it refers to the time to an episode defined by the analyst.
i) First is a keyword to designate the type of data design. Set this to EventTime.
ii) Next (after a delimiting | symbol) is a keyword to indicate when a visit should be deemed to occur for purposes of creating summary analysis variables and for determining if a visit is in-range. Set this to either ADMDATE or SEPDATE. Note that when the <episodes> occur is specified separately for each EDV episode type on the EPIOCCUR argument.

iii) Next (after a delimiting | symbol) is a combination of HPOI variables (delimited by the * symbol) to uniquely define the experimental unit in which to group and count episodes. For example, _HPOIDD_Prov*Person*POI_Dup uniquely identifies persons and _HPOIDD_Prov*HOSP_No uniquely identifies hospitals.
iv) Next (after a delimiting | symbol) is a keyword to indicate the time unit in which to record the time to next episode. This should be set to either Days or Weeks.

The output dataset will contain one record per experimental unit. Special variables will be added to contain the date the at-risk time for each episode type began for that experimental unit, and the time to first episode of each type starting from those points. The dates the at-risk time begins for each episode type for an experimental unit is the beginning of the valid episode date range specified on the DATERANGE argument. As with the output count datasets, the special variables will have the same base name as the episode but with a prefix affixed to the front. For each episode variable name EDVi defined in EDVNAME, _FstDateRsk_EDVi, _EventDate_EDVi, (_EventDays_EDVi or _EventWeeks_EDVi) and _Censored_EDVi are created to contain the first date the experimental unit is at risk for the episode named EDVi, the date and event time (in days or weeks) of the first EDVi type episode starting from the first date at risk, and a censoring indicator for whether the at risk time ended in an episode or the end of the at-risk time with no observed episode (right-censoring) (0=event, 1=censored). The end of the observation time is determined as the end of the valid DATERANGE argument for that episode. _EventDays_EDVi is calculated as _EventDays_EDVi=_EventDate_EDVi-_FstDateRsk_EDVi+1 day, while _EventWeeks_EDVi is calculated as _EventWeeks_EDVi=_EventDays_EDVi/7. Also included on the output dataset <only> when the experimental unit is person defined by _HPOIDD_Prov*Person*POI_Dup will be: _NumEDV_EDVi will hold the number of EDVs of the type EDVi in the event (first) EDVi episode (or missing if there is no event EDVi episode). _NumAllV_EDVi will hold the total number of visits in the event (first) EDVi episode (or missing if there is no event EDVi episode). _DistinctDays_EDVi will contain the number of distinct days in hospital within the EDVi event episode (or missing if there is no EDVi event), not counting any day more than once in the case of overlapping visits. _OvercountDays_EDVi will contain the number of days in hospital within the EDVi event (or missing if there is no EDVi type episode) allowing overlapping days to be counted more than once. This could be useful for example in a study involving health care costs billed.

Application:
Event-time data can be used to analyze hazard rates. Methods include non-parametric analyses such as Kaplan-Meier (perhaps stratified and analyzed in part with the log rank test) or life table, semi-parametric methods such as the Cox proportional hazards model, and fully parametric regression models such as exponential or Weibull regression.

Example:
```
%HPOIDD_Episode(...,EventTime|AdmDate|_HPOIDD_Prov*Hosp_No|Days,...);
```

Episode level data
You only need to specify the keyword EpisodeLevel.

The "experimental unit" in this case is the episode itself; that is, the output data will have one record per episode. Since episodes are defined within persons, there may be multiple records per person.

The output dataset will contain all episodes, one episode per record. There will be special variables added to the output dataset. _EpisodeType will contain the episode name (from EDVNAME). _EpiDate will contain the episode date (determined by the date specified in EPIOCCUR). _NumALLV will contain the number of visits counted towards that episode. _NumEDV will contain the number of EDVs of the type indicated in _EpisodeType counted within that episode. _DistinctDays will contains the number of distinct days in hospital within that episode, not counting any day more than once in the case of overlapping visits. _OvercountDays will contain the number of days in hospital within that episode, allowing overlapping days to be counted more than once. This could be useful for example in a study involving health care costs billed. The person identifier variables _HPOIDD_Prov Person POI_Dup are automatically kept on the output dataset, since episodes are always formed within persons.

Application:
Episode level data can be used to compare the characteristics of episodes (e.g., total length of stay, average age at onset, sex and more) between hospitals, health regions, provinces, or even on person level variables. Various methods including multiple linear regression or other GLMs such as binary or ordinal logistic regression, GEE versions of these GLMs—for correlated data when there are multiple records on a given experimental unit—and more.

Example:
```
%HPOIDD_Episode(...,EpisodeLevel,...);
```

Episode array data

You only need to specify the keyword EpisodeArray.
Under the episode array data design, each output record corresponds to a person (the experimental unit under this design). The analyst can specify one or many episode definitions, and each of these will contribute to zero or more episodes for a given person in an overall array of episodes of mixed type, ordered by episode date. Persons with zero total episodes are excluded from the output dataset.

The output dataset will contain all persons, one person per record. There will be special variables added to the output dataset. Where MAXEPISODES is the maximum number of episodes for a single person of all episode kinds combined, _GrandMaxEpisodes will equal MAXEPISODES, while _NumEpisodes will equal the number of episodes for each person. Other that that, all the variables available with EpisodeLevel data including those resulting from the AVARLIST argument will also be available, but with an integer from 1 to MAXEPISODES appended. For example, _NumALLV1-_NumALLVMAXEPISODES will contain the number of visits counted towards each episode of care in the array of episodes. The person identifier variables _HPOIDD_Prov Person POI_Dup are automatically kept on the output dataset, since episodes are always formed within persons.

Warning:

In output datasets under the EpisodeArray data design, due to the flat file format there are potentially many missing or blank variables created for all subjects with less than the global maximum number of episodes. If a small number of subjects have a very large number of episodes, **this can cause the output dataset to be very many times larger than the output dataset from a corresponding call with the EpisodeLevel data design**. Therefore if either EpisodeLevel or EpisodeArray can be used equally effectively for a given analysis, it is recommended to use the EpisodeLevel design.

Application:
This will be useful for analyses involving questions amongst several different episode types simultaneously, such as (generically) length of time to an episode of type A from the end of an episode of type B. One example could be time to death from an episode of AMI (acute myocardial infarction), or time to pacemaker implantation from AMI and then time to death following pacemaker. One other example could be time to death from an episode of AMI (acute myocardial infarction), or competing risks between time to pacemaker implantation from AMI and time to death from AMI.

Example:
```
%HPOIDD_Episode(...,EpisodeArray,...);
```

AVarList

Set AVARLIST to _NOAVAR is there are no analysis summary variables to request. Otherwise set AVARLIST to a space-delimited list of words defining which summaries of variables out of those available on the HPOIDD_BigData dataset and/or the user-defined variables to be generated on the output analysis dataset. Each word should start with a prefix indicating the summary, then contain a keyword indicating the subgroup on which to base the summary, then the episode type name from EDVNAME, and finally the base name of the variable to summarize.

For all data designs, available summary prefixes are:
Table, _Ntot_, _Nnmiss_, _NDist_, _Sum_, _Mean_, _SD_, _SE_, _Min_, _Max_, and _PXX_ where XX is an integer between 1 and 99. Respectively, these generate variables containing the following summary measures of a variable of interest: a full comma-delimited list of values and frequencies up to a maximum of 32767 characters before truncation in the form VALUE:FREQUENCY, the number of values (distinct, missing or not) of the variable of interest, the number of non-missing values (distinct or not) of the variable of interest, the number of distinct and non-missing values of the variable of interest, and the sum, mean, standard deviation, standard error, minimum, maximum and XXth percentile of the variable of interest.

Two additional summary prefixes are available only when the experimental unit is person, defined by _HPOIDD_Prov*Person*POI_Dup. These are _FirstV_ and _LastV_, respectively generating variables containing the values from the first encountered visit and the last encountered visit in the visit subgroup of interest (see next paragraph). These prefix keywords are available in the following situations: when the data design is Count and the experimental unit is _HPOIDD_Prov*Person*POI_Dup, when the data design is EventTime and the experimental unit is _HPOIDD_Prov*Person*POI_Dup, when the data design is EpisodeLevel since episodes

are the experimental unit and episodes are defined within persons, and when the data design is EpisodeArray since the experimental unit is person defined by _HPOIDD_Prov*Person*POI_Dup.

Note:
- In the following description, a visit is considered "in-range" if it occurs in the corresponding date range. An episode is considered "in-range" if it occurs in the corresponding date range. When a visit occurs is specified on the DATADESIGN argument. When episodes occur is specified for each episode on the EPIOCCUR argument.

For the Count data design, available visit subgroup keywords are:
AllVisIR_, EpiVisIR_, AllEDVIR_, EpiEDVIR_, EpiVis_ and EpiEDV_. These respectively mean that the summary of the variable on each output record (i.e., for each experimental unit) should be obtained among: all in-range visits (visits falling within DATERANGE), all in-range visits (whether or not they have EDV=1) that form part of an EDV episode, all in-range visits with EDV=1 (whether or not they form part of an EDV episode), all in-range visits with EDV=1 that also form part of an EDV episode, all visits (in-range or not, EDV=1 or not) that form part of an EDV episode (the EDV episode itself must be in range to be so defined), and all visits (in-range or not) with EDV=1 that form part of an EDV episode (which must itself be in range to be so defined).

For the EpisodeLevel data design, available visit subgroup keywords are:
EpiVis_ and EpiEDV_.
These respectively mean that the summary of the variable on each output record (i.e., for each episode) should be obtained among: all visits (in-range or not, EDV=1 or not) that form part of the EDV episode on that record (the EDV episode itself must be in range to be so defined), and all visits (in-range or not) with EDV=1 that form part of the EDV episode on that record (which must itself be in range to be so defined).

For the EpisodeArray data design, available visit subgroup keywords are:
EpiVis_ and EpiEDV_.
The summaries of the variable will be made on each episode in the array of episodes. The keywords listed above respectively mean to create the summary of the variable for each given episode in the array of episodes obtained among: all visits (in-range or not, EDV=1 or not) that form part of the EDV episode (the EDV episode itself must be in range to be so defined), and all visits (in-range or not) with EDV=1 that form part of the EDV episode (which must itself be in range to be so defined).

For the EventTime data design, available visit subgroup keywords are:
EpiVisIR_, EpiEDVIR_, EpiVis_ and EpiEDV_.
These respectively mean that the summary of the variable on each experimental unit should be obtained among: all in-range visits (whether or not they have EDV=1) that form part of the event EDV episode, all in-range visits with EDV=1 that also form part of the event EDV episode, all visits (in-range or not, EDV=1 or not) that form part of the event EDV episode (the EDV episode itself must be in range to be so defined), and all visits (in-range or not) with EDV=1 that form part of the event EDV episode (which must itself be in range to be so defined).

To obtain baseline summaries as explanatory variables, you can define a dummy episode called Baseline in EDVNAME with a statement in your SAS code setting Baseline=1 only if ADMDATE or SEPDATE or (ADMDATE+SEPDATE)/2 (or something else) is in a range of dates that is your baseline period (this might be from 1900.01.01 if no minimum for example, to the day before the date range of the outcome episode). Then use a washout time of 9999 weeks to combine all visits in the baseline range into one "baseline" episode, use a washout type of EDVS_IRSEP to ensure that the baseline episode ends prior to observation time for the outcome variable, and finally use the visit subgroup keyword EpiVisIR_ or EpiEDVIR_ or EpiVis_ or EpiEDV_ (in this case all will give same value due to how the baseline episode was set up). You can use the _LastV prefix (if your experimental unit is person defined by _HPOIDD_Prov*Person*POI_Dup) to give you the last known baseline value before the observation time for the outcome begins, or perhaps the _Mean prefix (not requiring the experimental unit to be person) which would give the average baseline value of the explanatory variable for the experimental unit before observation time.

Next, <only> if the episode data design is Count or EventTime, is the episode type name from EDVNAME followed by an _ symbol.

Finally the base variable name to summarize out of those available on the HPOIDD_BigData dataset and/or the user-defined variables to be generated on the output analysis dataset. Here you can specify either a variable you defined in your SAS code, or one of the available variables in HPOIDD_BigData datasets. For a list of available variables in HPOIDD_BigData datasets, refer to the HPOIDD user's guide or submit the following statement in SAS:

```
%HPOIDD_BigData_List_AvailVars;
```

Warnings:
i) A few of the available variable names in HPOIDD_BigData datasets are too long to have some of the prefixes attached. In those cases, you must create a shorter variable name by including lines in your SAS code like this for character variables:

```
 length Short $256.;
 Short=Longervariablename;
```

or this for numeric variables:

```
 Short=Longervariablename;
```

ii) If you intend to reference your summary analysis variables using array declarations for example under the EpisodeArray design, you must ensure that the base name does not end in an integer or an error will result. For example, the statement: "array _FirstV_EpiVis_Age_By5{7};" will throw an error. If you need to produce a summary of a variable whose name ends in an integer (e.g., Age_By5) and you need to use array statements on that summary variable, then you must first rename the variable as a user-defined variable into something that does not end in an integer (e.g., Age_By5T) and then request summary analysis variables of that new variable.

For example, suppose episode names OA and AnyVisit were defined on EDVNAME, user-defined variables Age and Comorbid were created in the input SAS code, and the data design was set to Count. Then the following argument might be used for AVARLIST:

```
%HPOIDD_Episode(...,
         _Mean_AllVisIR_OA_Age _Mean_EpiVisIR_OA_Age
         _p25_EpiVisIR_OA_comorbid _p50_EpiVisIR_oa_comorbid
```

```
_p75_EpiVisIR_oa_comorbid _Table_EpiVisIR_AnyVisit_Age_By5,...);
```

5.3 The episodes algorithm

SASCODEPATH, EDVNAME, EDVMIN, EPIOCCUR, DATERANGE, WASHTIME, WASHTYPE, WASHCOMP, SWVNAME, SWVLOGIC and DATADESIGN are nuanced concepts in HPOIDD. It is important that the analyst understand exactly what HPOIDD does given a particular definition.

1. The SAS code contained in the SASCODEPATH file will be run on the input HPOIDD_BigData generated dataset. Here, subsets of interest can be defined if needed. Code can refer to the variables available on HPOIDD_BigData generated datasets. After this code is run, every record will be flagged either 0 or 1 for each EDV variable listed on EDVNAME and each SWV variable listed on SWVNAME.

2. For each person defined by an _HPOIDD_Prov*Person*POI_Dup combination, the macro will enumerate a full array of all that person's visits ordered by ADMDATE and then SEPDATE in the event that ADMDATE is tied.

3. If optional SWVs are provided, then each required and precluding EDV-SWV relationship defined in SWVLOGIC is checked against the EDV in that relationship. Precluding relationships defined in SWVLOGIC that are satisfied cause the EDV to lose its EDV status (EDV variable is set to 0). Required relationships defined in SWVLOGIC that are not satisfied also cause an EDV to lose its EDV status. Recall that EDV-SWV relationships can be based on the relative occurrence of these events. Recall also that a visit cannot be counted as being its own SWV. For example, if during a visit that includes a diagnosis of osteoarthritis a person also has a bone density scan, and bone density scan occurring (midpoint of visit) within 2 weeks precludes a visit with an OA diagnosis from defining an episode, the bone scan during the same visit as the diagnosis of OA does not preclude that same visit's EDV status. However, that visit's scan might preclude a different visit's EDV status. If however you want a bone density scan during a visit to preclude the EDV status on that same visit also, it's very simple: build that condition into the SAS code that defines the EDV variable.

4. For each ith EDV variable named in EDVNAME: if the corresponding keyword in washout type list WASHTYPE is EDVS then any EDV=0 visits are stripped from the array (only in the copy of the array for that EDV). If the corresponding keyword is ALLVS then all visits (EDV or not) are retained in the copy of the array for that EDV at this step. If the corresponding keyword in washout type list WASHTYPE is EDVS_IRSEP then any EDV=0 visits or those with SEPDATE not in the date range of the episode definition are stripped from the array (only in the copy of the array for that EDV). If the corresponding keyword is ALLVS_IRSEP then all visits (EDV or not) that have in-range SEPDATE are retained in the copy of the array for that EDV at this step. If the corresponding keyword in washout type list WASHTYPE is EDVS_IRADM then any EDV=0 visits or those with ADMDATE not in the data range of the episode definition are stripped from the array (only in the copy of the array for that EDV). If the corresponding keyword is ALLVS_IRADM then all visits (EDV or not) that have in-range ADMDATE are retained in the copy of the array for that EDV at this step.

5. For each ith EDV variable named in EDVNAME: the program then cycles through each visit in the remaining array for that EDV from earliest to latest (ordered by ADMDATE and then SEPDATE). If the two visits occur close enough together then the second visit is concatenated onto the episode containing the first visit, and the comparison moves one visit over to the 2nd vs. 3rd visit. E.g., if there are four visits labeled A, B, C and D, A and B could be joined into an episode, then B and C deemed close enough for C to join the episode that B is in, but then D might occur far enough after C that D is not joined but starts its own new episode. This would leave two episodes, the first made up of visits A, B and C, and D comprising the other episode. The ith comparison made in WASHCOMP indicates from what dates in each of the current and next visit the dates should be considered when forming the ith EDV episode type, for example separation date in the current visit (e.g., visit A) might commonly be compared with admission date in the next visit (e.g., visit B). Then the time difference of date of B minus date of A is compared to the ith washout time in WASHTIME. If the current to next visits are at least as far apart as the time indicated in WASHTIME, then they are considered distinct visits and the second is not joined onto the episode that the first visit is part of. Recall that any negative comparison time (for example if the current visit's separation date occurs after the next visit's admission or midpoint date and the corresponding ith WASHCOMP comparison is SEP TO ADM or SEP TO MID) will be treated as 0 only if the NEGATIVES_TO_0 option is used in that entry in WASHCOMP (corresponding ith WASHCOMP comparison "SEP TO ADM NEGATIVES_TO_0" or "SEP TO MID NEGATIVES_TO_0"). If the NEGATIVES_TO_0 option is used, then having a WASHTIME of 0 days will mean that two visits with a negative time comparison will still be considered distinct. The result of this step is 0 or more episodes per _HPOIDD_Prov*Person*POI_Dup combination, each of which contains 0 or more EDVs.

6. All episodes that contain less than the minimum required number of EDVs for that episode type (specified in EDVMIN) are dropped from the array of episodes, and what remains is 0 or more episodes each of which contain at least the minimum required number of EDVs for that episode type.

7. Occurrence dates for each episode are calculated depending on the specifications in the EPIOCCUR argument, as either admission, separation or midpoint of either the first visit, last visit, middle visit, first EDV, last EDV or middle EDV.

8. The next step is for the macro to run through the episodes remaining and exclude those episodes that do not occur (based on EPIOCCUR) inside the valid date range specified for that episode type in DATERANGE.

9. This dataset with an array of episodes for each person is processed into an analysis dataset via the DATADESIGN specifications.

10. Finally, the summary variables specified in AVARLIST are generated for each record in the output dataset per the AVARLIST specifications.

Notes:
i) For a list of available variables in HPOIDD_BigData datasets, refer to the HPOIDD user's guide or submit the following statement in SAS:

```
%HPOIDD_BigData_List_AvailVars;
```

ii) SAS code may include many lines of code.
iii) For detailed examples consult the following chapters.

5.4 Sub-setting the input dataset for testing purposes

HPOIDD_BigData datasets can be very large (several gigabytes). As a result, it can take HPOIDD_Episode a long time (possibly hours depending on the speed of your computer) to prepare an analysis dataset. Although many errors and inconsistencies are caught early in a run during preliminary checks, some cannot be found until farther into the run, and it can be very frustrating to wait an hour or more only to encounter an error message and have to reconfigure your macro call and start over. A simple solution is to test any new call to HPOIDD_BigData on a subset of the HPOIDD_BigData dataset, e.g., a single province, or for a smaller subset a single province and gender or age group, etc. After preparing an analysis dataset on a subset of the HPOIDD_BigData data, **carefully read over the OutText data dictionary** to ensure that the final dataset and episode definitions are in the correct form, before proceeding with a lengthy run on full HPOIDD_BigData data.

This is very easy to do. For example, to include only records on females from British Columbia, as the first line of your SAS code file, put the statement:

```
if _HPOIDD_Prov eq 59 and Sex eq "2";
```

Of course if the variable PROV is part of your experimental unit, you may want to subset by something else for the test run.

6. Statistical analysis of HPOIDD_Episode and HPOIDD_BigData data

6.1 Overview

The HPOIDD_BigData dataset can be analyzed whenever the analysis involves analyzing visits, as the HPOIDD_BigData dataset has one record per visit. The custom designed output dataset from the HPOIDD_Episode macro can be analyzed in various ways, depending on how the data were defined.

HPOIDD_Episode can produce count, episode level, episode array or event-time data for user-defined episode types. In general, count data can be used in various analyses including incidence rates, odds ratios, Poisson regression or other generalized linear models (GLMs)[6] such as binary or ordinal logistic regression, repeated measures ANOVA if counts are high enough for a normal approximation, generalized estimating equations (GEE)[12] versions of the aforementioned GLMs—for correlated data when there are multiple records on a given experimental unit—and more. Event-time data can be used in the analysis of hazard rates. Methods include non-parametric analyses such as Kaplan-Meier (perhaps stratified and analyzed in part with the log rank test) or life table, semi-parametric methods such as the Cox proportional hazards model, and fully parametric regression models such as exponential or Weibull regression. The summary variables specified on the AVARLIST argument can be analyzed as well. For example, under the **EpisodeLevel** data design, a linear regression model for distinct number days of stay in hospital (_DistinctDays) during episodes of acute myocardial infarction (AMI) in Ontario could be fit against Hosp_No over all available data years, or in a repeated measures ANOVA where each experimental unit (e.g., episode) has an analysis summary table type variable on the HPOI variable Hosp_No (e.g., _Table_EpiVis_Hosp_No), as well as _EpiDate for determining the year of each episode.

Cross-sectional models run on **Count** data can be used to investigate the present association between different variables measured at the same time. For example, crude or stratified odds ratios or logistic regression could be used to investigate how whether or not a subject is hospitalized for a tabulating diagnosis of OA in a given fiscal year (i.e., _Count_OA>0) might relate to the subject's age (in the case of a 2 by 2 table, whether the subject is >60 years old or not) halfway through that year. Such an analysis using only HPOI data would have a reference population consisting of all hospitalized persons admitted in the year of interest (who were discharged within the available years); the representativeness of such an analysis would have to be carefully considered. Such models could be run on person, hospital, or even region or province-level experimental units.

An alternative analysis representative of the general Canadian population could be performed if the data were linked to national survey data (e.g., National Population Health Survey (NPHS) or the Canadian Community Health Survey (CCHS)). Such data could be analyzed as **retrospective**, **case-control** data. Cases would be all those hospitalized with a tabulating diagnosis of, for example, a particular form of cancer in a given fiscal or calendar year (who were discharged within the available years). Controls, sampled from the national survey data, would be those persons who were not hospitalized for this reason *or not discharged for such a*

hospitalization within the available years. Case/control analyses could be used to study the difference between the distribution of an exposure of interest (e.g., percent aged 15 years and older with less than grade 9 education in the enumeration area of the hospital (cases) or respondent (controls)) in cases versus controls. Such an analysis would usually be limited to the variables available in HPOI data, since cases mostly would not appear in national survey data. The only exception to this is if the case records could be linked to other sources of data that contained variables available also on the controls. Case-control analyses utilizing the odds ratio as a measure of association are useful in part because they can be inverted to estimate causal effects of exposure on the probability of caseness. Examples of methods for *unmatched* case-control data producing odds ratios are the crude odds ratio from a 2 by 2 table and, if there are covariates of interest, unconditional logistic regression. Corresponding methods for matched (e.g., on province, age and/or sex) case-control data are the stratified Mantel-Haenszel odds ratio, and conditional logistic regression. Such models could be run on person, hospital, or even region or province-level experimental units—the middle two provided that sufficient information exists to assign national survey respondents (controls) to hospital or regional experimental units.

Longitudinal models can be used to investigate the relationship between variables measured repeatedly over time. For example, the number or rate of episodes of a particular viral infection in patients (tabulating diagnosis or not) could be compared between hospitals in different regions depending on disease trends of interest perhaps following an experimental vaccination campaign that was only done in selected health regions, and rates of infection could be repeatedly measured each calendar or fiscal year. Depending on how long the vaccination takes to become biologically effective, a repeated measures analysis such as repeated measures ANOVA, or a generalized linear model (GLM) adjusting for correlated responses within experimental units via generalized estimating equations (GEE), could be informative. Such models could be run on person, hospital, or even region or province-level experimental units, probably one of the latter three in the example of a vaccination program.

The longitudinal models described above are done so in the context of **prospective**, **cohort** data, which effectively begins with a population or sample and follows it over time. Cross-sectional methods can be used on such data if the outcome is only measured once at the end of the study period, compared to an exposure that is randomly assigned or measured at the beginning, perhaps along with covariates, and what happens in between is not of interest (sometimes this is due to limitations in the data). HPOI data linked with national survey data could be treated as cohort data as well. Rather than analyzing a case-control dataset of all hospitalizations for a given episode type attached to a set of controls from the survey data, one could restrict one's analyses to some subset of a national survey dataset. Survey weights (final and replicate weights for variance estimation) would ensure that the results represented the Canadian population. Another obvious advantage is that there would be many (hundreds) more variables to analyze. The main disadvantage is that if the outcome is rare, cohort studies are less efficient than case-control studies, due to small cell sizes. Complex regression models (e.g., logistic regression) of rare outcomes may therefore only be possible on case-control data, but it's a trade off as the variables in such a dataset are generally limited to HPOI variables.

Most of the examples above implicitly describe analyses of HPOIDD_Episode output datasets from the **Count** data design either directly or converted as a mean into experimental unit-level

prevalence (e.g., per health region). Event-time data (available under the data design type **EventTime**) otherwise known as lifetime data from HPOIDD_Episode can also be analyzed in the context of cohort data. Various event-time models could be fit on prospective EventTime data. Subjects in a cohort never hospitalized for a given episode type would represent right-censored event times. Linkage to longitudinal surveys such as the NPHS would be especially useful in such analyses, allowing for sophisticated "repeated measures" event-time models such as the Cox proportional hazards model (PHM) with time-varying covariates. Event-time models can be fit to HPOIDD_Episode output datasets most straightforwardly from the **EventTime** and **EpisodeArray** data designs, but could also be fit using EpisodeLevel or even Count data if done with caution.

Caveats of HPOI data

There is an important caveat in modeling with HPOI data. This is the fact that HPOI hospital separation records are not generated until a patient is discharged or transferred. Subjects remaining in hospital for a lengthy time and persons who are not hospitalized at all are therefore invisible to the analyst. This can affect what conclusions can be drawn from various statistical models, from modeling counts or continuous outcomes, to modeling of event-time data.

As a direct example, suppose in an EventTime design the event of interest is hospital discharge from a starting time of admission (i.e., length of stay is the outcome). Then only observed event times exist in the data—there are no censored event times recorded. So if the analyst is analyzing how event times relate to some factor, and one factor level sometimes experiences much longer event times but the width of the data window is too narrow to observe the longer event times, then the event times in the two groups can appear in the data to be closer than they really are, and the analysis will find a washed out result that is biased towards the null hypothesis. As a solution, an analysis of length of stay for example might "...study the effect of sex and age at admission on the length of first stay in hospital **that was discharged in fiscal year 2000/1**." In other words, to deal with the selection bias this analysis would redirect the focus of the study onto those who were discharged in a particular year. This is perfectly legitimate, however it is important to understand this limitation and to report it along with any findings.

Another direct example is an EventTime design in which admission to hospital for a particular reason is the event of interest. Then those who are not in HPOI data at all will be invisible to the analysis, when again they should be included as right-censored observations. A solution to this second example is to link HPOI data to national survey or census data, so that those who are not in HPOI data at all can be included in the analysis dataset. Note however that for experimental units such as hospital, health region or province, analyses *are* representative of Canada generally and this problem of invisible units does not apply, because such data are generally a census of those units when all available HPOI data are used. That is, all hospitals in Canada should be found in the HPOI data, but not all *persons*.

There are other, less obvious situations. For example, perhaps an analyst has linked EventTime data to prospective cohort survey data and is analyzing time to admission for some condition amongst only those subjects in the cohort. Once again, only those who experience the episode but are also discharged or at least transferred mid-episode soon enough to appear in the HPOI data will have recorded events. Longer initial hospital stays will not appear as events at all, and

what are possibly the most serious events early on in a person's observation period will be treated as non-events, with right censoring at the far right end of the data.

The good news, however, is that since longer event times are in general missed more than short ones, any difference between groups is likely if anything to be somewhat washed out. Therefore this bias may be perceived (in many cases) as a **generally conservative bias** favoring the null hypothesis, and if necessary can be reported as such.

In addition to understanding and reporting this selection bias along with study findings, an analyst may also want to estimate (at least the approximate) magnitude of the bias on a particular result. To do this for a particular analysis, one could perform a **sensitivity analysis**, successively removing HPOI data years to shorten the window of observation and recording what happens to the direction and size of the model parameter estimates.

Another approach to fitting event time models to data with this issue is to restrict the window of analysis to a smaller sub-period of the available time with sufficient lag time following the observation window in order to ensure that no patients currently in hospital are invisible to the analysis. However, the trade-off is that you are losing data.

6.2 Example of preparing the HPOIDD_BigData dataset

Before we can use the HPOIDD_Episode macro to create analysis-ready datasets from HPOI data, we must combine all available years of HPOI data into one larger SAS dataset via the HPOIDD_BigData macro. In this example, the following HPOI datasets are available in the folder h:\HPOI data\.

CAN datasets:
CAN9293, CAN9394, CAN9495, CAN9596, CAN9697, CAN9798, CAN9899, CAN9900, CAN0001, CAN0102, CAN0203, CAN0304

Diagnosis datasets:
Diagnosis0102, Diagnosis0203, Diagnosis0304

Intervention datasets:
Intervention0102, Intervention0203, Intervention0304

We combine these into one SAS dataset called HPOIDD_BigData92to03 stored in a local folder for faster analysis via the following HPOIDD_BigData macro call. We use options to change blank POI_Dup values to 0's, and to change single-character Person values to blank (assuming they are errors).

```
LIBNAME netlib v9 "h:\HPOI data\";
LIBNAME loclib v9 "h:\HPOI data\";
%HPOIDD_BigData(netlib,_AllData,loclib.HPOIDD_BigData92to03,
        CHANGE_POI_DUP_BLANKS_TO_ZEROS,
        CHANGE_PERSON_CH1_TO_BLANK);
```

After this run has completed, we can analyze the loclib.HPOIDD_BigData92to03 dataset in any number of analyses. We will use this dataset as our HPOIDD_BigData dataset in the following examples of the HPOIDD_Episode macro. However, in situations where the experimental unit is visit, there is no need to run HPOIDD_Episode; the HPOIDD_BigData dataset can be analyzed directly.

6.3 Poisson regression

Poisson regression is a generalized linear model (GLM). The response variable is an integer count variable, and what is modeled is the (generally non-integer) expected count given a set of covariates. The GLM can be specified as follows.

The *random component* is the conditional distribution of Y_i. In Poisson regression, it may be no surprise that the Poisson distribution is assumed. Let $\eta = \mathbf{x}'\beta$ be the *systematic component*, or linear predictor. Where μ is the conditional expectation of Y_i given the covariates, let $g(\mu) = \log(\mu)$ be the *link function* that links the expected value of the response with the linear predictor η. In the Poisson model, the log link is the "canonical" link, which is related to what is called the canonical form or expression of the Poisson distribution.

Example 1 – GLM fit on Count data in independent hospital-level data records

The analyst wants to use a Poisson regression model to regress the expected number of tabulating OA diagnoses admitted per hospital (the experimental unit) between June 15, 1994 and March 31, 1999 on some explanatory variables and other covariates. The HPOI sample is presumed to represent the relevant population of hospitals.

(The analyst understands that the only admissions counted in the data are those that are subsequently discharged within the available years of data.)

Warning:
The ICD-9, ICD-9-CM, ICD-10, CCP and/or CCI codes presented in this example are for structural and syntactical illustration only and may not be accurate codes for the diseases and/or procedures in this example.

The call to the HPOIDD_Episode macro is set up as follows.

- The input dataset is specified via INDAT as loclib.HPOIDD_BigData92to03 (created above in section *6.2 Example of preparing the HPOIDD_BigData dataset*).
- The output SAS dataset is specified in OUTDAT to be saved in SASLIB1.HospCountOA.
- The output text readme file for the output dataset is specified in OUTTEXT to be saved in d:\bin\HospCountOA.txt.
- The SAS code defining episodes is defined in SASCODEPATH to be located in d:\bin\HospCountOA_SAScode.txt.
- The single episode type of interest is named OA in the EDVNAME argument and this 0-1 EDV variable is defined in the SAS code under ICD-9, ICD-9-CM and ICD-10 specifications.

- A single episode-defining visit (EDV) variable value of 1 is sufficient to count the episode, specified via EDVMIN.
- Episode dates are considered to be the first episode EDV admission date, specified via EPIOCCUR.
- Only episodes occurring between June 15, 1994 and March 31, 1999 are counted, specified via DATERANGE.
- For each person, starting from the 2nd visit and proceeding forward through all their visits (sorted by ADMDATE SEPDATE), each current visit is joined with the previous visit's episode if (comparison date of the current visit)-(comparison date of the previous visit)<1 week (specified via WASHTIME). This includes situations where the difference between dates is negative, which can happen when there are partially or fully overlapping visits and the current visit's comparison date (in this example we use ADMDATE) is before the previous visit's comparison date (in this example we use SEPDATE). The previous and current visit's comparison dates are specified via WASHCOMP. We allow all in-range visit types (not just EDVs) to contribute to episodes (specified via WASHTYPE).
- Special washout visit (SWV) settings specify that potential OA visits with ADMDATE between -5 days after (i.e., 5 days before) and 52 weeks after the SEPDATE of a visit with a *first* diagnosis code of Felty's syndrome are precluded from being OA EDVs. The SWV name Felty is specified via SWVNAME, and this 0-1 SWV variable is defined in the SAS code under ICD-9, ICD-9-CM and ICD-10 specifications. The SWV logic is specified via SWVLOGIC.
- DATADESIGN specifies the data design. The Count data design is specified. For the purposes of calculating in-range summary analysis variables specified on AVARLIST, and for determining whether a visit is in-range or not, a visit is deemed to occur on ADMDATE. The experimental unit is hospital defined by _HPOIDD_Prov*Hosp_No. The time unit in which to group and count episodes is specified as TotalTime since this analysis has one period of interest (it is not a per-year analysis for example).
- In addition to defining the EDV and SWV variables, in the SAS code there is also a user-defined comorbidity indicator variable defined indicating the presence on the discharge form of osteoporosis: ICD-9/ICD-9-CM code 733.0 "Osteoporosis"; ICD-10 code M80 "Osteoporosis with pathological fracture"; or ICD-10 code M81 "Osteoporosis without pathological fracture". This indicator variable is named UDef_OP, is numeric and takes integer values 0 or 1. The mean (prevalence) of this indicator per hospital will be included in the regression model as a potential confounder.
- AVARLIST specifies the analysis summary variables to include on the output dataset. _Mean_AllVisIR_OA_UDef_OP will contain (for each record's experimental unit in the output dataset) the mean value of UDef_OP amongst all visits occurring in-range for the OA episode (whether OA=1 for the visit or not) where the valid date range for OA episodes was defined earlier in DATERANGE.
- The analyst has also performed an external analysis using Census data in order to produce an auxiliary SAS dataset named sasliba.AuxInfo with extra variables to be linked to each hospital in the HPOIDD_Episode analysis-ready output dataset. The extra variables in this example are categorical median income in 1994 (MedianIncomeGroup94) and median BMI in 1994 (MeanBMI94) in the service area of each hospital. The auxiliary dataset also includes _HPOIDD_Prov and Hosp_No in order to be linkable to the output

HPOIDD_Episode dataset whose experimental units are defined by _HPOIDD_Prov*Hosp_No combination.

```
%HPOIDD_Episode(
     loclib.HPOIDD_BigData92to03,
     SASLIB1.HospCountOA,
     d:\bin\HospCountOA.txt,
     d:\bin\HospCountOA_SAScode.txt,
     OA,
     1,
     FEDV_ADM,
     1994.06.15-1999.03.31,
     1 weeks,
     AllVs,
     SEP to ADM,
     Felty,
     Felty Precludes OA EDVADM-SWVSEP from -5 days to 52 weeks,
     Count|ADMDATE|_HPOIDD_Prov*Hosp_No|TotalTime,
     _Mean_AllVisIR_OA_UDef_OP
     );
```

The contents of d:\bin\HospCountOA_SAScode.txt are:

```
*** User-defined 0-1 comorbidity variable UDef_OP;
UDef_OP9=0;
UDef_OP9CM=0;
UDef_OP10=0;
do i=1 to _HPOIDD_DIAGNOSIS_ALEN;
 UDef_OP9=UDef_OP9+(substr(scan(DIAG_ICD9_CODE{i},1,' '),1,5) eq "733.0");
 UDef_OP9CM=UDef_OP9CM+(substr(scan(DIAG_CM_CODE{i},1,' '),1,5) eq "733.0");
 UDef_OP10=UDef_OP10+(upcase(substr(scan(DIAG_ICD10_CODE{i},1,' '),1,3)) in
("M80" "M81"));
end;
UDef_OP=(UDef_OP9+UDef_OP9CM+UDef_OP10 gt 0);

*** The OA EDV variable;
OA9=0;
OA9CM=0;
OA10=0;
do i=1 to _HPOIDD_DIAGNOSIS_ALEN; * Any OA diagnosis is counted, first or
not;
 OA9=OA9+(substr(scan(DIAG_ICD9_CODE{i},1,' '),1,5) eq "714.3");
 OA9CM=OA9CM+(substr(scan(DIAG_CM_CODE{i},1,' '),1,6) eq "714.31");
 OA10=OA10+(upcase(substr(scan(DIAG_ICD10_CODE{i},1,' '),1,5)) in ("M08.3"));
end;
OA=(OA9+OA9CM+OA10 gt 0);

*** The special washout visit (SWV) variable Felty;
Felty9=0;
Felty9CM=0;
Felty10=0;
do i=1 to 1; * Felty diagnosis is only counted if it is the first diagnosis;
 Felty9=Felty9+(substr(scan(DIAG_ICD9_CODE{i},1,' '),1,5) eq "714.1");
 Felty9CM=Felty9CM+(substr(scan(DIAG_CM_CODE{i},1,' '),1,5) eq "714.1");
```

```
 Felty10=Felty10+(upcase(substr(scan(DIAG_ICD10_CODE{i},1,' '),1,5)) in
("M05.0"));
end;
Felty=(Felty9+Felty9CM+Felty10 gt 0);
```

The output dataset will contain the following variables.

- The experimental unit identifiers, in this example _HPOIDD_Prov and Hosp_No.
- The variables _Count_OA, _PersAtRisk_OA, _PTAtRisk_OA, _RecStaDate_OA and _RecEndDate_OA produced automatically when the data design type Count is specified.
- Whatever special analysis summary variables are requested, in this case _Mean_AllVisIR_OA_UDef_OP.

The following SAS code shows how to link these data, and perform Poisson regression on them.

```
/*** SASLIB1.HospCountOA should already be sorted by
     _HPOIDD_Prov*Hosp_No ***/
proc sort data=sasliba.AuxInfo; by _HPOIDD_Prov Hosp_No; run;
data usedat;
 merge SASLIB1.HospCountOA sasliba.AuxInfo;
 by _HPOIDD_Prov Hosp_No;
 run;

data usedat;
 set usedat;
 log_PersonYearsAtRisk_OA=log(_PTAtRisk_OA/365.25);
 run;

proc genmod data=usedat;
 title1 "Example of Poisson regression";
 class MedianIncomeGroup94(param=ref);
 model _Count_OA=MedianIncomeGroup94 MeanBMI94 _Mean_AllVisIR_OA_UDef_OP
  / dist=poisson link=log offset=log_PersonYearsAtRisk_OA;
 run;
```

The class statement tells SAS that MedianIncomeGroup94 is a categorical variable and to use reference cell coding (also known as treatment contrasts). That is when one category of the variable is treated as the reference group and there is a coefficient for each of the remaining categories. The model statement indicates that the data are Poisson distributed count data and to use the log link (the canonical link for the Poisson model). The offset option accounts for the fact that total person time at risk differs between hospitals. Having an offset of log of person years at risk for each hospital means that count per person year is modeled. This is due to count for a hospital being count per person year times the number of person years of data for that hospital, therefore log count is log count per person year plus log of person years of data for that hospital. The model coefficients from this model estimate the effect on the log count (per person year) due to each covariate.

Example 2 – GLM with GEE fit on Count data in repeated measures hospital-level data

There are various reasons why an analyst might want to fit the Poisson regression in a repeated measures model. Such count data might be repeatedly measured each fiscal year. In that case the

call to HPOIDD_Episode would differ from the previous call in the specification of the data design. The DATADESIGN argument would be set to Count|ADMDATE|_HPOIDD_Prov*Hosp_No|FiscalYear. DATERANGE could be changed to 1994.04.01-1999.03.31 to contain complete fiscal years (or the first partial year would have a lower expected total count), however since we're using log of person time as an offset and hence are modeling expected count per person year at risk, this is not necessary. The output dataset would then contain one record per hospital per fiscal year, and contain the variable _FiscalYear to record the fiscal year for each record as a 6-digit number. The auxiliary dataset would ideally now have a record for each _HPOIDD_Prov*Hosp_No*_FiscalYear combination with extra variables MedianIncomeGroup and MeanBMI now applicable per year. The analysis could account for the correlated data likely to result from having multiple records on the same hospitals by generalized estimating equations (GEE) modeling. Merging of the data and Poisson regression with GEE could be run by the following calls in SAS. (Code that is different than the previous example is shown in bold.)

```
/*** SASLIB1.HospCountOA should already be sorted by
     _HPOIDD_Prov*Hosp_No*_FiscalYear ***/
proc sort data=sasliba.AuxInfo; by _HPOIDD_Prov Hosp_No _FiscalYear; run;
data usedat;
 merge SASLIB1.HospCountOA sasliba.AuxInfo;
 by _HPOIDD_Prov Hosp_No _FiscalYear;
 run;

data usedat;
 set usedat;
 log_PersonYearsAtRisk_OA=log(_PTAtRisk_OA/365.25);
 run;

proc genmod data=usedat;
 title1 "Example of Poisson regression with GEE";
 class MedianIncomeGroup(param=ref);
 model _Count_OA=MedianIncomeGroup MeanBMI _Mean_AllVisIR_OA_UDef_OP
  / dist=poisson link=log offset=log_PersonYearsAtRisk_OA;
 repeated subject=_HPOIDD_Prov*Hosp_No / type=exch;
 run;
```

In the above example, the *working correlation matrix* is specified as type "exchangeable", which means that a single shared correlation should be estimated for the off diagonals of the matrix of correlations between the repeated measurements on the same hospital. This is the most parsimonious working correlation matrix, but may not be adequate in some situations. For more details about this and other working correlation types, see the SAS 9.1.3 online help.

6.4 Logistic regression

The logistic regression model is another GLM[7]. The response variable can be ordinal (ordered categorical) but is more commonly a simple binary (0,1). What is modeled is the cumulative (or reverse cumulative) probability of different levels of the response. In the commonly specified binary model with "descending" SAS option, the point probability of the highest ordered category (generally, 1) is modeled, given the set of covariates. This GLM can be specified as follows.

The *random component* is the conditional distribution of Y_i. In binary logistic regression, the Bernoulli, also called the binomial(1) distribution is assumed. Let $\eta = \mathbf{x}'\beta$ be the *systematic component*, or linear predictor. Where π is the conditional probability that Y_i=1 given the covariates, let $g(\pi) = \text{logit}(\pi) = \log(\pi/(1-\pi))$ be the *link function* that links the expected value of the response with the linear predictor η. In the Bernoulli model, the logit link is the "canonical" link, which is related to what is called the canonical form or expression of the binomial(1) distribution.

Example 1 – GLM fit on Count data in independent prospective person-level cohort data linked to NPHS

The analyst wants to use a logistic regression model to regress the probability of a person (the experimental unit) being admitted to hospital with a first diagnosis of OA between June 15, 1994 and March 31, 1999 on some explanatory variables and other covariates. In this example, the HPOIDD_Episode dataset is to be linked to national survey data, and the analysis dataset is restricted to subjects sampled in the national survey. This is done to produce prospective cohort data representative of the 1994 Canadian population. Advantages are the large number of variables available on the national survey data to use in the modeling, and that persons not appearing in HPOI are included in the dataset. The downside is that much of the HPOI data must be discarded because they do not belong to subjects in the survey sample.

(The analyst understands that the only admissions counted in the data are those that are subsequently discharged within the available years of data.)

Warning:
The ICD-9, ICD-9-CM, ICD-10, CCP and/or CCI codes presented in this example are for structural and syntactical illustration only and may not be accurate codes for the diseases and/or procedures in this example.

The call to the HPOIDD_Episode macro is set up as follows.

- The input dataset is specified via INDAT as loclib.HPOIDD_BigData92to03 (created above in section *6.2 Example of preparing the HPOIDD_BigData dataset*).
- The output SAS dataset is specified in OUTDAT to be saved in SASLIB1.PersonCountOA.
- The output text readme file for the output dataset is specified in OUTTEXT to be saved in d:\bin\PersonCountOA.txt.
- The SAS code defining episodes is defined in SASCODEPATH to be located in d:\bin\PersonCountOA_SAScode.txt.
- The single episode type of interest is named OA in the EDVNAME argument and this 0-1 EDV variable is defined in the SAS code under ICD-9, ICD-9-CM and ICD-10 specifications.
- 2 episode-defining visits (EDVs) with variable value of 1 are required to count the episode, specified via EDVMIN.
- Episode date is considered to be the last visit's (EDV or not) separation date in the episode, specified via EPIOCCUR.

- Only episodes occurring between June 15, 1994 and March 31, 1999 are counted, specified via DATERANGE.
- For each person, starting from the 2nd visit and proceeding forward through all their visits (sorted by ADMDATE SEPDATE), each current visit is joined with the previous visit's episode if (comparison date of the current visit)-(comparison date of the previous visit)<1 week (specified via WASHTIME). This includes situations where the difference between dates is negative, which can happen when there are partially or fully overlapping visits and the current visit's comparison date (in this example we use ADMDATE) is before the previous visit's comparison date (in this example we use SEPDATE). The previous and current visit's comparison dates are specified via WASHCOMP. We allow all in-range visit types (not just EDVs) to contribute to episodes (specified via WASHTYPE).
- There are no special washout visit (SWV) variables specified. This is indicated via SWVNAME and SWVLOGIC set to _NOSWV.
- DATADESIGN specifies the data design. The Count data design is specified, and the Count per person will be converted to a 0-1 indicator prior to analysis. For the purposes of calculating in-range summary analysis variables specified on AVARLIST, and for determining whether a visit is in-range or not, a visit is deemed to occur on ADMDATE. The experimental unit is person defined by _HPOIDD_Prov*Person*POI_Dup. The time unit in which to group and count episodes is specified as TotalTime since this analysis has one period of interest (it is not a per-year analysis for example).
- There are no user-defined variables in this example.
- AVARLIST specifies the analysis summary variables to include on the output dataset. _Mean_AllVisIR_OA_UDef_OP will contain (for each record's experimental unit in the output dataset) the mean value of UDef_OP amongst all visits occurring in-range for the OA episode (whether OA=1 for the visit or not) where the valid date range for OA episodes was defined earlier in DATERANGE.
- The analyst has enabled a linkage between HPOI and the National Population Health Survey (NPHS), and has created the linkage variables _HPOIDD_Prov, Person and POI_Dup which match those variables in HPOIDD_BigData. The NPHS-based auxiliary SAS dataset is named sasliba.AuxInfo. The extra variables the analyst has derived from the NPHS for this example are type of smoker (daily, occasional, not at all), type of drinker (regular, occasional, former, never), sex, and age, all measured at baseline in 1994 (age is calculated on January 1, 1994 from the DOB recorded in the survey data). These variables are named SmokerType, DrinkerType, Sex and Age1994. The dataset also includes _HPOIDD_Prov, Person and POI_Dup in order to be linkable to the output HPOIDD_Episode dataset whose experimental units are defined by _HPOIDD_Prov*Person*POI_Dup. The survey dataset also includes the survey weight named FWGT and a set of 1000 replicate weights for complex survey variance estimation, in this example bootstrap weights, named BSW1-BSW1000.
- AVARLIST specifies the analysis summary variables to include on the output dataset. _FirstV_AllVisIR_OA_SEX and _FirstV_AllVisIR_OA_BTHDATE will be used in data integrity checks; in linked data these should match the corresponding quantities on the survey data. That is, sex and age on the survey data should match these variables amongst those persons who are in both the national survey data and the HPOIDD_Episode data records. The reason for specifying a visit subgroup keyword of AllVisIR is to check the

data amongst *all* persons with visits in-range for OA appearing on both data sources, not just those who experience episodes of OA.

```
%HPOIDD_Episode(
     loclib.HPOIDD_BigData92to03,
     SASLIB1.PersonCountOA,
     d:\bin\PersonCountOA.txt,
     d:\bin\PersonCountOA_SAScode.txt,
     OA,
     2,
     LV_SEP,
     1994.06.15-1999.03.31,
     1 weeks,
     AllVs,
     SEP to ADM,
     _NoSWV,
     _NoSWV,
     Count|ADMDATE|_HPOIDD_Prov*Person*POI_Dup|TotalTime,
     _FirstV_AllVisIR_OA_SEX _FirstV_AllVisIR_OA_BTHDATE
     );
```

The contents of d:\bin\PersonCountOA_SAScode.txt are:

```
*** The OA EDV variable;
OA9=0;
OA9CM=0;
OA10=0;
do i=1 to _HPOIDD_DIAGNOSIS_ALEN; * Any OA diagnosis is counted, first or
not;
 OA9=OA9+(substr(scan(DIAG_ICD9_CODE{i},1,' '),1,5) eq "714.3");
 OA9CM=OA9CM+(substr(scan(DIAG_CM_CODE{i},1,' '),1,6) eq "714.31");
 OA10=OA10+(upcase(substr(scan(DIAG_ICD10_CODE{i},1,' '),1,5)) in ("M08.3"));
end;
OA=(OA9+OA9CM+OA10 gt 0);
```

The output dataset will contain the following variables.

- The experimental unit identifiers, in this example _HPOIDD_Prov, Person and POI_Dup.
- The variables _Count_OA, _PersAtRisk_OA, _PTAtRisk_OA, _RecStaDate_OA and _RecEndDate_OA produced automatically when the data design type Count is specified.
- Whatever special analysis summary variables are requested, in this case _FirstV_AllVisIR_OA_SEX and _FirstV_AllVisIR_OA_BTHDATE.

The following SAS code shows how to link these data, and perform weighted logistic regression on them.

```
/*** SASLIB1.HospCountOA should already be sorted by
     _HPOIDD_Prov*Person*POI_Dup ***/
proc sort data=sasliba.AuxInfo; by _HPOIDD_Prov Person POI_Dup; run;
data usedat;
 merge SASLIB1.PersonCountOA sasliba.AuxInfo;
 by _HPOIDD_Prov Person POI_Dup;
 run;
```

```
data usedat suspect;
 set usedat;
 /*** Create binary outcome from count variable ***/
 bOA=(_Count_OA gt 0);
 output usedat;

 /*** Data integrity checks. Dataset Work.Suspect should contain 0 records.
      Sex should match between HPOI and the survey data, as should age
      at least to within one year.
 ***/
 if _FirstV_AllVisIR_OA_BDY_OLD eq 0 then _FirstV_AllVisIR_OA_BDY_OLD=1;
 Check_Age1994=round(mdy(1,1,1994)-_FirstV_AllVisIR_OA_BTHDATE)/365.25;
 if abs(Check_Age1994-Age1994) gt 1 or
    _FirstV_AllVisIR_OA_Sex ne Sex then output suspect;
 run;

proc genmod descending data=usedat;
 title1 "Example of Logistic regression";
 class SmokerType DrinkerType Sex / param=ref;
 model bOA=SmokerType DrinkerType Sex Age1994
            / dist=binomial link=logit;
 weight fwgt;
 run;
```

The class statement tells SAS that SmokerType, DrinkerType and Sex are categorical variables and to use reference cell coding. The model statement indicates that the data are binomial(1) distributed data and to use the logit link (the canonical link for the binomial(1) model). The model coefficients from this model estimate the effect on the log odds of an OA episode due to each covariate adjusted for the others.

Example 2 – GLM with GEE fit on Count data in repeated measures prospective person-level cohort data linked to NPHS

Suppose it were desired that the data be repeatedly measured each fiscal year. In that case the call to HPOIDD would differ from the previous call in the specification of the data design. The DATADESIGN argument would be set to Count|ADMDATE|_HPOIDD_Prov*Person*POI_Dup|FiscalYear. DATERANGE could be changed to 1994.04.01-1999.03.31 to contain complete fiscal years or the probability of an OA episode would be lower in the first partial year, or else some adjustment for person-time at risk could be made. The output HPOIDD_Episode dataset would contain one record per person per fiscal year, and contain the output variable _FiscalYear to record the fiscal year for each record as a 6-digit number. The analysis could account for the correlated data likely to result from this by GEE modeling. Merging of the data and Logistic regression with GEE could be run on the output dataset by the following calls in SAS. Age is now calculated on April 1st of the fiscal year for each record instead of just in 1994. (Code that is different than the previous call is in bold.)

```
/*** SASLIB1.HospCountOA should already be sorted by
     _HPOIDD_Prov*Person*POI_Dup ***/
proc sort data=sasliba.AuxInfo; by _HPOIDD_Prov Person POI_Dup; run;
data usedat;
 merge SASLIB1.PersonCountOA sasliba.AuxInfo;
 by _HPOIDD_Prov Person POI_Dup;
 run;
```

```
data usedat suspect;
 set usedat;
 /*** Create binary outcome from count variable ***/
 bOA=(_Count_OA gt 0);
 output usedat;

 /*** Data integrity checks. Dataset Work.Suspect should contain 0 records.
      Sex should match between HPOI and the survey data, as should age
      at least to within one year.
 ***/
 if _FirstV_AllVisIR_OA_BDY_OLD eq 0 then _FirstV_AllVisIR_OA_BDY_OLD=1;
 Age=round(mdy(4,1,substr(_FiscalYear,1,4))-
           _FirstV_AllVisIR_OA_BTHDATE)/365.25;
 if _FirstV_AllVisIR_OA_Sex ne Sex then output suspect;
 run;

proc genmod descending data=usedat;
 title1 "Example of Logistic regression with GEE";
 class SmokerType DrinkerType Sex / param=ref;
 model bOA=SmokerType DrinkerType Sex Age
            / dist=binomial link=logit;
 repeated subject=_HPOIDD_Prov*Person*POI_Dup / type=MDEP(3);
 run;
```

In the above example, the working correlation matrix is specified as type "m-dependent" with m=3 (fiscal years), which means that observations more than 3 years apart are assumed to be independent, but correlations are estimated for each of 1, 2 and 3 year separations of data records within subjects. For more details about this and other working correlation types, see the SAS 9.1.3 online help.

Example 3 – GLM fit on per visit HPOIDD_BigData data

The analyst wants to use a logistic regression model to regress the probability of a hospital separation being discharged dead versus alive, between June 15, 1994 and March 31, 1999 on age, sex, province and acute versus non-acute hospital. The experimental unit therefore is visit. In these situations there is no need to run HPOIDD_Episode; the HPOIDD_BigData dataset can be analyzed directly. This analysis is performed on unlinked data.

The logistic regression model can be run on the HPOIDD_BigData dataset by the following calls in SAS.

```
data usedat (keep=bDead Age Acute Sex _HPOIDD_Prov);
 set loclib.HPOIDD_BigData92to03;
 if SEPDATE ge mdy(6,15,1994) and SEPDATE le mdy(3,31,1999);
 bDead=(DIS_OLD ne 1);
 Age=(SEPDATE-BTHDATE)/365.25;
 run;

proc genmod descending data=usedat;
 title1 "Example of Logistic regression";
 class Acute Sex Prov / param=ref;
 model bDead=Acute Sex Prov Age / dist=binomial link=logit;
 run;
```

The class statement tells SAS that Acute, Sex and Prov are categorical variables and to use reference cell coding. The model statement indicates that the data are binomial(1) distributed data and to use the logit link (the canonical link for the binomial(1) model). The model coefficients from this model estimate the effect on the log odds of being discharged dead (amongst all in-scope hospital separations) due to each covariate adjusted for the others.

6.5 Linear regression and repeated measures ANOVA

Linear regression[8] is a standard method of analyzing continuous response data with respect to categorical and/or continuous fixed explanatory variables. The assumptions are independent and identically distributed normal error terms (residuals). Independence of error terms usually requires that the data contain no more than one record per subject.

Repeated measures ANOVA is a method of analyzing continuous data that is measured repeatedly on the same subjects over time[9]. Error terms from observations on the same subject tend to be correlated. The repeated measures ANOVA analyzes the set of outcomes per subject as a response vector. Multivariate normality of error terms is assumed between repeated measures within subjects. The assumption of equally spaced observation times is made, but the common "multivariate" approach to repeated measures ANOVA, which we take in the following example, is thought to be somewhat robust to violations of this assumption.

Example 1 – Multiple linear regression fit to summary analysis variable for days of stay in independent EpisodeLevel data

The analyst wants to use a linear regression model to study the effect of sex and age at first admission per episode on the length of an episode in hospital that ended in fiscal year 2000/1 (the episode may have begun before the year). New visits are considered part of the previous episode if ADMDATE is less than 60 days after the last SEPDATE. The analyst wants to include in the analysis only HPOIDD_BigData records generated by residents of the reporting province (RES_FLAG will be used in the SAS code to define this subgroup). The analyst also wants to restrict the analysis to subjects whose *first* visit to hospital in the available HPOI data was a short stay of 1 week or less. The regression will be done by province.

The call to the HPOIDD_Episode macro is set up as follows.

- The input dataset is specified via INDAT as loclib.HPOIDD_BigData92to03 (created above in section *6.2 Example of preparing the HPOIDD_BigData dataset*).
- The output SAS dataset is specified in OUTDAT to be saved in SASLIB1.EpisodeLevelLengthStay.
- The output text readme file for the output dataset is specified in OUTTEXT to be saved in d:\bin\EpisodeLevelLengthStay.txt.
- The SAS code defining the subgroup of interest and the episodes is defined in SASCODEPATH to be located in d:\bin\EpisodeLevelLengthStay_SAScode.txt.
- The single episode type of interest is named Stay in the EDVNAME argument and this 0-1 EDV variable is defined in the SAS code to be a dummy variable identically 1.
- At least 1 episode-defining visit (EDV) with variable value of Visit=1 is required to count the episode, specified via EDVMIN.

- Episode date is considered to be the last visit's (EDV or not) separation date in the episode, specified via EPIOCCUR.
- Only episodes (last visit separations) occurring between April 1, 2000 and March 31, 2001 are counted, specified via DATERANGE.
- We set WASHTIME=60 days, WASHTYPE=ALLVS and WASCOMP=SEP TO ADM.
- There are no special washout visit (SWV) variables specified. This is indicated via SWVNAME and SWVLOGIC set to _NOSWV.
- DATADESIGN specifies the data design. The EpisodeLevel data design is specified. The experimental unit under the EpisodeLevel data design is episode, which automatically occurs within person defined by _HPOIDD_Prov*Person*POI_Dup. So there will be one or more records per person in the output data.
- There are no user-defined variables in this example.
- AVARLIST specifies the analysis summary variables to include on the output dataset. _LastV_EpiVis_Stay_BTHDATE, _LastV_EpiVis_Stay_Sex and _LastV_EpiVis_Stay_Prov will contain (for each person in the output dataset) the birth date, sex and province code assessed at separation from the last in-range visit where the valid date range for Visit episodes was defined earlier in DATERANGE. (We set Prov equal to _HPOIDD_Prov in the SAS code since Prov is not on HPOIDD_BigData datasets.) We also request _FirstV_EpiVis_Stay_ADMDATE which is the admission date of the first visit in the episode. Our main analysis variables, _DistinctDays and _OvercountDays will be retained automatically since this design has a subspace of person as the experimental unit.

```
%HPOIDD_Episode(
     loclib.HPOIDD_BigData92to03,
     SASLIB1.EpisodeLevelStay,
     d:\bin\EpisodeLevelStay.txt,
     d:\bin\EpisodeLevelStay_SAScode.txt,
     Stay,
     1,
     LV_SEP,
     2000.04.01-2001.03.31,
     60 days,
     ALLVS,
     SEP to ADM,
     _NoSWV,
     _NoSWV,
     EpisodeLevel,
     _LastV_EpiVis_Stay_BTHDATE _LastV_EpiVis_Stay_Sex
     _LastV_EpiVis_Stay_Prov _FirstV_EpiVis_Stay_ADMDATE
     );
```

The contents of d:\bin\EpisodeLevelStay_SAScode.txt are:

```
*** Only include visits by residents of the reporting province;
*** and only include those whose first visit to hospital in the;
*** available HPOI data was a short stay of 1 week or less.;
/* Remember the by statement automatically used in this data step:
   by _HPOIDD_PROV PERSON POI_DUP ADMDATE SEPDATE
      _HPOIDD_DATA_YR _HPOIDD_SEP_NUM;
```

```
   and remember that person is defined by:
   _HPOIDD_PROV*PERSON*POI_DUP
*/
retain bFirstVisShort;
if first._HPOIDD_PROV or first.PERSON or first.POI_DUP then do;
 if (SEPDATE-ADMDATE) le 7 then bFirstVisShort=1;
 else bFirstVisShort=0;
end;
if RES_FLAG eq 0 and bFirstVisShort;

*** Set a new numeric variable Prov to _HPOIDD_Prov;
Prov=_HPOIDD_Prov;

*** We use a dummy EDV in this example to potentially count all visits;
Stay=1;
```

The output dataset will contain the following variables.

- In EpisodeLevel data there is one record per person-episode. To identify person there are the person identifiers _HPOIDD_Prov, Person and POI_Dup. To identify episode there are the variables _EpisodeType and _EpiDate.
- The variables _NumALLV, _NumEDV, _DistinctDays and _OvercountDays are also produced automatically when the data design type EpisodeLevel is specified.
- Whatever special analysis summary variables are requested, in this case _LastV_EpiVis_Stay_BTHDATE, _LastV_EpiVis_Stay_Sex, _LastV_EpiVis_Stay_Prov and _FirstV_EpiVis_Stay_ADMDATE.

The linear regression model can be run on the output dataset by the following calls in SAS. We use Proc GLM because it supports categorical explanatory variables without the analyst having to manually code dummy variables.

```
data usedat (keep=AgeEpiStart Sex Prov);
 set saslib1.EpisodeLevelStay;
 AgeEpiStart=(_FirstV_EpiVis_Stay_ADMDATE-
              _LastV_EpiVis_Stay_BTHDATE)/365.25;
 if _LastV_EpiVis_Stay_Sex in (1 2) then Sex=_LastV_EpiVis_Stay_Sex;
 else Sex=.;
 Prov=_LastV_EpiVis_Stay_Prov;
 run;
proc sort data=usedat; by Prov; run;

proc glm data=usedat;
 title1 "Example of Linear Regression on EpisodeLevel data";
 class Sex;
 model _DistinctDays=AgeEpiStart Sex / solution;
 by Prov;
 run;
 quit;
```

In this situation we analyze _DistinctDays to avoid multiple-counting of days in the case of overlapping stays. However, another study (perhaps a study involving health care costs billed) might look at _OvercountDays instead, which does allow multiple-counting of the same day when there are overlapping days. The call to Proc GLM specifies that the total days in hospital

during the episodes will be regressed on the explanatory variables AgeEpiStart and Sex, with Sex a categorical variable (Proc GLM uses reference cell coding). The "solution" option requests the regression coefficients. The regression is done by province. For more details, see the SAS 9.1.3 online help.

Example 2 – Repeated measures ANOVA fit to Count data in linked hospital-level data measured repeatedly over several fiscal years

The analyst wants to use a repeated measures ANOVA to study the effect of an experimental hospital-wide policy intervention on counts of OA episodes per hospital per fiscal year. It is assumed that admission counts are big enough to justify the assumption of normality so that (repeated measures) ANOVA can be used. The five fiscal years from 1994/5 to 1998/9 are of interest. There is an auxiliary dataset containing the hospital identifier variables _HPOIDD_Prov, Hosp_No, the fiscal year indicator _FiscalYear, plus a 0-1 indicator for the intervention, in this example an experimental policy implemented in a random sample of hospitals for fiscal years starting in 1994/5.

(The analyst understands that the only admissions counted in the data are those that are subsequently discharged within the available years of data.)

Warning:
The ICD-9, ICD-9-CM, ICD-10, CCP and/or CCI codes presented in this example are for structural and syntactical illustration only and may not be accurate codes for the diseases and/or procedures in this example.

The call to the HPOIDD_Episode macro is set up as follows.

- The input dataset is specified via INDAT as loclib.HPOIDD_BigData92to03 (created above in section *6.2 Example of preparing the HPOIDD_BigData dataset*).
- The output SAS dataset is specified in OUTDAT to be saved in SASLIB1.CountHospOA.
- The output text readme file for the output dataset is specified in OUTTEXT to be saved in d:\bin\CountHospOA.txt.
- The SAS code defining episodes is defined in SASCODEPATH to be located in d:\bin\CountHospOA_SAScode.txt.
- The single episode type of interest is named OA in the EDVNAME argument and this 0-1 EDV variable is defined in the SAS code under ICD-9, ICD-9-CM and ICD-10 specifications.
- 1 episode-defining visit (EDV) with variable value of 1 is required to count the episode, specified via EDVMIN.
- Episode date is considered to be the first EDV visit's ADMDATE in the episode, specified via EPIOCCUR.
- Only episodes occurring between April 1, 1994 and March 31, 1998 are counted (so that each fiscal year in the output dataset is a complete fiscal year), specified via DATERANGE.
- For each person, starting from the 2nd visit and proceeding forward through all their visits (sorted by ADMDATE SEPDATE), each current visit is joined with the previous visit's episode if (comparison date of the current visit)-(comparison date of the previous visit)<1

week (specified via WASHTIME). This includes situations where the difference between dates is negative, which can happen when there are partially or fully overlapping visits and the current visit's comparison date (in this example we use ADMDATE) is before the previous visit's comparison date (in this example we use SEPDATE). The previous and current visit's comparison dates are specified via WASHCOMP. We allow all in-range visit types (not just EDVs) to contribute to episodes (specified via WASHTYPE).

- There are no special washout visit (SWV) variables specified. This is indicated via SWVNAME and SWVLOGIC set to _NOSWV.
- DATADESIGN specifies the data design. The Count data design is specified. For the purposes of calculating in-range summary analysis variables specified on AVARLIST, and for determining whether a visit is in-range or not, a visit is deemed to occur on ADMDATE. The experimental unit is hospital defined by _HPOIDD_Prov*Hosp_No. The time unit in which to group and count episodes is specified as FiscalYear.
- In addition to defining the EDV and SWV variables, in the SAS code there is also a user-defined comorbidity indicator variable defined indicating the presence on the discharge form of osteoporosis: ICD-9/ICD-9-CM code 733.0 "Osteoporosis"; ICD-10 code M80 "Osteoporosis with pathological fracture"; or ICD-10 code M81 "Osteoporosis without pathological fracture". This indicator variable is named UDef_OP, is numeric and takes integer values 0 or 1. The mean (prevalence) of this indicator amongst all in-range visits (EDV or not) per hospital-fiscal year will be included in the regression model as a potential confounder.
- AVARLIST specifies the analysis summary variables to include on the output dataset. _Mean_AllVisIR_OA_UDef_OP will contain (for each record's experimental unit and fiscal year in the output dataset) the mean value of UDef_OP amongst all visits occurring in-range for the OA episode (whether OA=1 for the visit or not) where the valid date range for OA episodes was defined earlier in DATERANGE.
- The analyst has performed linkage between HPOI and the auxiliary dataset described above, named sasliba.AuxInfo. The variable the analyst has put on this dataset in addition to _HPOIDD_Prov, Hosp_No and _FiscalYear used to identify hospital and fiscal year, is bNewPolicy94, a 0/1 indicator for whether the experimental policy was in effect in the hospital starting in the 1994/5 fiscal year.

```
%HPOIDD_Episode(
     loclib.HPOIDD_BigData92to03,
     SASLIB1.CountHospOA,
     d:\bin\CountHospOA.txt,
     d:\bin\CountHospOA_SAScode.txt,
     OA,
     1,
     FEDV_ADM,
     1994.04.01-1998.03.31,
     1 weeks,
     AllVs,
     SEP to ADM,
     _NoSWV,
     _NoSWV,
     Count|ADMDATE|_HPOIDD_Prov*Hosp_No|FiscalYear,
     _Mean_AllVisIR_OA_UDef_OP
     );
```

The contents of d:\bin\HospCountOA_SAScode.txt are:

```
*** User-defined 0-1 comorbidity variable UDef_OP;
UDef_OP9=0;
UDef_OP9CM=0;
UDef_OP10=0;
do i=1 to _HPOIDD_DIAGNOSIS_ALEN;
 UDef_OP9=UDef_OP9+(substr(scan(DIAG_ICD9_CODE{i},1,' '),1,5) eq "733.0");
 UDef_OP9CM=UDef_OP9CM+(substr(scan(DIAG_CM_CODE{i},1,' '),1,5) eq "733.0");
 UDef_OP10=UDef_OP10+(upcase(substr(scan(DIAG_ICD10_CODE{i},1,' '),1,3)) in
("M80" "M81"));
end;
UDef_OP=(UDef_OP9+UDef_OP9CM+UDef_OP10 gt 0);

*** The OA EDV variable;
OA9=0;
OA9CM=0;
OA10=0;
do i=1 to _HPOIDD_DIAGNOSIS_ALEN; * Any OA diagnosis is counted, first or
not;
 OA9=OA9+(substr(scan(DIAG_ICD9_CODE{i},1,' '),1,5) eq "714.3");
 OA9CM=OA9CM+(substr(scan(DIAG_CM_CODE{i},1,' '),1,6) eq "714.31");
 OA10=OA10+(upcase(substr(scan(DIAG_ICD10_CODE{i},1,' '),1,5)) in ("M08.3"));
end;
OA=(OA9+OA9CM+OA10 gt 0);
```

The output dataset will contain the following variables.

- The experimental unit identifiers, in this example _HPOIDD_Prov and Hosp_No.
- The variables _Count_OA, _PersAtRisk_OA, _PTAtRisk_OA, _RecStaDate_OA and _RecEndDate_OA produced automatically when the data design type Count is specified.
- Whatever special analysis summary variables are requested, in this case _Mean_AllVisIR_OA_UDef_OP.

The following SAS code shows how to link these data, and perform repeated measures ANOVA on them. Note that the comorbidity covariate the analyst puts in the model is, for each hospital, the average prevalence of the comorbidity indicator for that health region amongst all visits to that hospital within each fiscal year, averaged over all fiscal years to give one value per hospital. That is, the comorbidity covariate is the average value of _Mean_AllVisIR_OA_UDef_OP over time for each hospital.

```
/*** SASLIB1.HospCountOA should already be sorted by
     _HPOIDD_Prov*Hosp_No*_FiscalYear ***/
proc sort data=sasliba.AuxInfo; by _HPOIDD_Prov Hosp_No _FiscalYear; run;
data usedat;
 merge SASLIB1.HospCountOA sasliba.AuxInfo;
 by _HPOIDD_Prov Hosp_No _FiscalYear;
 run;

*** Put repeated measures data into vector form;
data usedat
  (keep=_HPOIDD_Prov Hosp_No
        CntOA199495 CntOA199596 CntOA199697 CntOA199798 CntOA199899
```

```
         bNewPolicy94 HospMean_UDef_OP);
 set usedat;
 by _HPOIDD_Prov Hosp_No _FiscalYear;
 retain CntOA199495 CntOA199596 CntOA199697 CntOA199798 CntOA199899
        HospMean_UDef_OP _numrec;
 if first._HPOIDD_Prov or first.Hosp_No then do;
  CntOA199495=.;
  CntOA199596=.;
  CntOA199697=.;
  CntOA199798=.;
  CntOA199899=.;
  _numrec=0;
  HospMean_UDef_OP=0;
 end;
 _numrec+1;
 HospMean_UDef_OP=HospMean_UDef_OP+HospMean_UDef_OP;
 if _FiscalYear eq 199495 then CntOA199495=_Count_OA;
 else if _FiscalYear eq 199596 then CntOA199596=_Count_OA;
 else if _FiscalYear eq 199697 then CntOA199697=_Count_OA;
 else if _FiscalYear eq 199798 then CntOA199798=_Count_OA;
 else if _FiscalYear eq 199899 then CntOA199899=_Count_OA;
 if last._HPOIDD_Prov or last.Hosp_No then do;
  HospMean_UDef_OP=HospMean_UDef_OP/_numrec;
  output usedat;
  _numrec=0;
 end;
 run;

proc glm data=usedat;
 title1 "Example of Repeated measures ANOVA";
 model CntOA199495 CntOA199596 CntOA199697 CntOA199798 CntOA199899=
       bNewPolicy94 HospMean_UDef_OP / solution;
 repeated Time 5 (0 1 2 3 4) / summary printe;
 run;
```

The model statement indicates that the 5 responses (one observed count from each of the five fiscal years) form a response vector to be regressed in a repeated measures ANOVA against the explanatory indicator variable bNewPolicy94 and the covariate HospMean_UDef_OP. The "solution" option requests the regression coefficient for bNewPolicy94 (adjusted for HospMean_UDef_OP), to estimate the effect of the new policy on the mean count. Of course SAS does not distinguish between explanatory variables and covariates, so a coefficient for HospMean_UDef_OP adjusted for bNewPolicy94 will also be shown. The "repeated" statement indicates that the repeated measures were taken once per year for five years. The "summary" option requests tests of the effects of each between-subject variable on the contrasts between each time point and the last. The "printe" option requests tests of "sphericity", a property of the error covariance matrix between time points within subjects that is an assumption in the multivariate tests in this model. The output from this analysis will include univariate tests for the effect of the new policy on rate of OA at each time point in the analysis, a regression coefficient to estimate that effect at each time point, multivariate tests for an overall effect (considering all time points), and more. For more details, see Montgomery[9] or the SAS 9.1.3 online help.

6.6 Retrospective case-control data

Case-control data is *retrospective* because the sample is stratified by the outcome (often a rare disease) after the fact. The case-control sample is usually taken this way to ensure that there is an adequate proportion of the sample who are cases, that is, who have a positive response. With HPOIDD, a case-control sample could be constructed by taking from the HPOI data all subjects who experience some (most likely rare) episode during some period of time. These are cases. Controls can then be selected from some external data source, for example population health surveys such as the NPHS or CCHS. If a sample weight variable exists on the data source for controls, an identically named variable would be placed on the case dataset, set identically to 1 to indicate that individual cases represent only themselves.

Analysis of case-control data can be done in a number of ways. The simplest method is the odds ratio from a 2 by 2 cross-table of case/control versus an indicator for exposure. The odds ratio is useful in case-control analyses because of its symmetry. The OR is a symmetric measure in that the effect of a dichotomous variable X being 1 on the odds of a dichotomous variable Y being 1 is also the effect of variable Y being 1 on the odds of a variable X being 1. This is easily shown:

$$\theta_{X\to Y} = \frac{P(Y=1\mid X=1)/(1-P(Y=1\mid X=1))}{P(Y=1\mid X=0)/(1-P(Y=1\mid X=0))}$$

$$= \frac{P(Y=1\mid X=1)/P(Y=0\mid X=1)}{P(Y=1\mid X=0)/P(Y=0\mid X=0)}$$

$$= \frac{\left[P(X=1\mid Y=1)\frac{P(Y=1)}{P(X=1)}\right]\Big/\left[P(X=1\mid Y=0)\frac{P(Y=0)}{P(X=1)}\right]}{\left[P(X=0\mid Y=1)\frac{P(Y=1)}{P(X=0)}\right]\Big/\left[P(X=0\mid Y=0)\frac{P(Y=0)}{P(X=0)}\right]}$$

$$= \frac{P(X=1\mid Y=1)/P(X=0\mid Y=1)}{P(X=1\mid Y=0)/P(X=0\mid Y=0)}$$

$$= \frac{P(X=1\mid Y=1)/(1-P(X=1\mid Y=1))}{P(X=1\mid Y=0)/(1-P(X=1\mid Y=0))}$$

$$= \theta_{Y\to X}$$

The odds ratio from a cross-table does have some drawbacks, such as a limited capacity for covariates. Some adjustment can be achieved via the stratified Mantel-Haenszel OR (MHOR).

Logistic regression can also be used to analyze case-control data. *Unconditional* logistic regression, described earlier, is suitable for analyzing *unmatched* case-control data. *Matched* case-control data should be analyzed with *conditional* logistic regression[7] (CLR). In CLR, the likelihood that is maximized is the conditional probability of the data given the unknown parameters, where conditioning is on the stratum totals and case counts which are sufficient statistics for the nuisance parameters (stratum specific intercepts), which are themselves therefore eliminated from the likelihood. Matching can be done in case-control studies to better control for confounders especially when the confounding variables have very different distributions in cases versus controls[10]. For example, cases may be much older on average than

controls, and age may also be related to the probability of exposure. Some common matching variables used in case-control studies are age, sex and geographic region.

In logistic regression, effect estimates are represented as odds ratios (by exponentiating the regression coefficients). In case-control data, the outcome (case/control status) is fixed since the data were collected on the basis of caseness, and the exposure is random. Therefore it is sensible that the probability of exposure should be regressed upon an indicator for caseness, and while odds ratios from such a regression will directly estimate the effect of caseness on the probability of exposure (not exactly what is needed), because the odds ratio is a symmetric estimator (as discussed above) the same OR also conveniently estimates the effect of exposure on the probability of caseness. But Hosmer and Lemeshow[7] show by repeated application of Bayes theorem that it is possible to invert the likelihood in such as way as to demonstrate equivalence to maximizing a likelihood with caseness treated directly as the response variable. This allows for the inclusion of other covariates in the model besides the exposure, and is what makes logistic regression so useful in a case-control analysis.

Example 1 – Unconditional logistic regression and unstratified odds ratio on Count data in unmatched person-level case-control data (using CCHS for controls)

Suppose the analyst wishes to study the effect of gender and age on the probability of admission to hospital for a tabulating diagnosis of rheumatoid lung disease in the province of Ontario. Since admission to hospital with this condition as the tabulating diagnosis is rare, it is decided to take a case-control approach. This example considering gender and age as the "exposure" is rather simplistic, but again, case-control analyses of HPOI data will often be limited to variables available both externally (on controls) *and* in the HPOI files (on cases). In situations where it is possible to link the case subset of HPOI persons to external data, variables other than those in the HPOI files could be studied as exposure variables in case-control analyses.

Rheumatoid lung disease is only defined under ICD-9-CM (714.81 "Rheumatoid lung") and ICD-10 (M05.1+ "Rheumatoid lung disease"). Therefore the analysis is restricted to discharges occurring in those years/provinces that use those coding systems, which happens to be 2001/2 and above except for Quebec. In this example the analyst is only using Ontario data. Depending on the length of stay, the admission dates could be in earlier years than 2001/2, but admissions are only counted if occurring in calendar years 2001 and 2002 to match the data collection period of the 2001/2 Canadian Community Health Survey (CCHS), as that is the source of controls for this analysis. Cases are all persons admitted to Ontario hospitals with a tabulating diagnosis of rheumatoid lung disease found in the HPOI data. Controls are all persons from the Ontario portion of the 2001/2 CCHS who are not cases.

(The analyst understands that the only admissions counted in the data are those that are subsequently discharged within the available years of data.)

Warning:
The ICD-9, ICD-9-CM, ICD-10, CCP and/or CCI codes presented in this example are for structural and syntactical illustration only and may not be accurate codes for the diseases and/or procedures in this example.

The call to the HPOIDD_Episode macro is set up as follows.

- The input dataset is specified via INDAT as loclib.HPOIDD_BigData92to03 (created above in section *6.2 Example of preparing the HPOIDD_BigData dataset*).
- The output SAS dataset is specified in OUTDAT to be saved in SASLIB1.CaseRheumLung.
- The output text readme file for the output dataset is specified in OUTTEXT to be saved in d:\bin\CaseRheumLung.txt.
- The SAS code defining the subgroup of interest and the episodes is defined in SASCODEPATH to be located in d:\bin\CaseRheumLung_SAScode.txt. In addition to defining the EDV variable, the SAS code also uses only the Ontario subset of the HPOIDD_BigData dataset.
- The single episode type of interest is named RLng in the EDVNAME argument and this 0-1 EDV variable is defined in the SAS code under ICD-9-CM and ICD-10.
- At least 1 episode-defining visit (EDV) with variable value of RLng=1 is required to count the episode, specified via EDVMIN.
- Episode date is considered to be the first EDV's admission date in the episode, specified via EPIOCCUR.
- Only episodes (ADMDATE of first EDV) occurring between January 1, 2000 and December 31, 2001 are counted, specified via DATERANGE.
- We set WASHTIME=9999 weeks, WASHTYPE=AllVs and WASHCOMP=SEP TO SEP NEGATIVES_TO_0 to ensure that if there are multiple in-range EDV visits, all are combined into one episode and only the first EDV is counted for each case.
- There are no special washout visit (SWV) variables specified. This is indicated via SWVNAME and SWVLOGIC set to _NOSWV.
- DATADESIGN specifies the data design. The Count data design is specified. Counts will be converted to an indicator variable pre-analysis. For the purposes of calculating in-range summary analysis variables specified on AVARLIST, and for determining whether a visit is in-range or not, a visit is deemed to occur on ADMDATE. The experimental unit is person defined by _HPOIDD_Prov*Person*POI_Dup. The time unit in which to group and count episodes is specified as TotalTime since this analysis has a single two-year period of interest (it is not a per-year analysis for example).
- The analyst has prepared an external data set with controls from the Ontario portion of the 2001/2 CCHS. The dataset is named sasliba.Controls. It contains the variable Sex and a derived variable Age1994 (assessed January 1, 1994), to correspond to those variables in HPOI. It contains the indicator Caseness set to 0 on all records (since these are controls). The dataset also includes the survey weight named FWGT from the CCHS sample weight, which records the number of persons represented by each control.
- There are no user-defined variables in this example.
- AVARLIST specifies the analysis summary variables to include on the output dataset. _FirstV_EpiEDVIR_RLng_BTHDATE and _FirstV_EpiEDVIR_RLng_Sex are (for each person in the output dataset) the birth date and sex measured at separation from the first in-range EDV in the episode where the valid date range for RLng episodes was defined earlier in DATERANGE.

```
%HPOIDD_Episode(
```

```
     loclib.HPOIDD_BigData92to03,
     SASLIB1.CaseRheumLung,
     d:\bin\CaseRheumLung.txt,
     d:\bin\CaseRheumLung_SAScode.txt,
     RLng,
     1,
     FEDV_ADM,
     2000.01.01-2001.12.31,
     9999 weeks,
     AllVs,
     SEP to SEP NEGATIVES_TO_0,
     _NoSWV,
     _NoSWV,
     Count|ADMDATE|_HPOIDD_Prov*Person*POI_Dup|TotalTime,
     _FirstV_EpiEDVIR_RLng_BTHDATE _FirstV_EpiEDVIR_RLng_Sex
     );
```

The contents of d:\bin\CaseRheumLung_SAScode.txt are:

```
*** Use only the Ontario subset of the HPOIDD_BigData dataset;
if _HPOIDD_Prov eq 35;

*** The RLng EDV variable;
RLng9CM=0;
RLng10=0;
* Only a tabulating, or first diagnosis of RLng is counted;
do i=1 to 1; *** Not i=1 to _HPOIDD_DIAGNOSIS_ALEN;
 RLng9CM=RLng9CM+(substr(scan(DIAG_CM_CODE{i},1,' '),1,6) eq "714.81");
 RLng10=RLng10+(upcase(substr(scan(DIAG_ICD10_CODE{i},1,' '),1,5)) in
("M05.1"));
end;
RLng=(RLng9CM+RLng10 gt 0);
```

The output dataset will contain the following variables.

- The experimental unit identifiers, in this example _HPOIDD_Prov, Person and POI_Dup.
- The variables _Count_RLng, _PersAtRisk_RLng, _PTAtRisk_RLng, _RecStaDate_RLng and _RecEndDate_RLng produced automatically when the data design type Count is specified.
- Whatever special analysis summary variables are requested, in this case _FirstV_EpiEDVIR_RLng_BTHDATE and _FirstV_EpiEDVIR_RLng_Sex.

The case-control dataset can be built from the HPOIDD_Episode output Case dataset and analyzed by the following calls in SAS. For illustrative purposes we use Proc Logistic instead of Proc Genmod as used earlier.

```
data cases (keep=fwgt Caseness Age1994 Sex) suspect;
 set saslib1.CaseRheumLung;
 /*** With a washout of 9999 weeks no-one should have more than one
      EDV visit ***/
 if _Count_RLng gt 1 then output suspect;

 /*** Keep only cases, and set weight to 1 since each case
      represents only himself or herself ***/
```

```
 if _Count_RLng eq 1;
 Caseness=1;
 fwgt=1;
 /*** Exposure variables ***/
 Sex=_FirstV_EpiEDVIR_RLng_Sex;
 if Sex eq 0 then Sex=.; * Set unknown sex to missing;
 Age1994=(mdy(1,1,1994)- _FirstV_EpiEDVIR_RLng_BTHDATE)/365.25;
 output cases;
 run;

data usedat; set cases sasliba.Controls; run;

proc logistic descending data=usedat;
 title1 "Example of Unconditional logistic regression ";
 title2 "On unmatched case-control data";
 class Sex / param=ref;
 model Caseness=Sex Age1994;
 weight fwgt;
 run;

proc freq data=usedat;
 title1 "Unstratified odds ratio for sex versus rheumatic lung";
 title2 "On unmatched case-control data";
 title3 "(Result may differ from above logistic analysis since;
 title4 "effect of sex is not adjusted for age group)";
 tables Sex*Caseness / measures;
 weight fwgt;
 run;
```

The class statement in the Proc Logistic call tells SAS that Sex is a categorical variable and to use reference cell coding. The model coefficients from this model estimate the effect on the log odds of an RLng episode of each exposure variable controlling for the other. The odds ratio estimated in the Proc Freq call can be used as a rough comparison. However, it should be noted that the estimated effect of sex in this call has not been adjusted for age, and therefore some difference may be expected.

Example 2 – Conditional logistic regression and stratified Mantel-Haenszel odds ratio on Count data in matched person-level case-control data (using CCHS for controls)

Suppose the analyst wishes to expand the analysis to all of Canada. There are some important considerations that must be made. First, ICD-9-CM and ICD-10 coding systems are not available in Quebec data in the years of interest. Even beyond that, province is a potential confounder, because age and sex distributions can differ between provinces, and the probability of a tabulating diagnosis of rheumatic lung disease will differ according to climate and air quality (which vary according to province). It is therefore decided that province should be a matching variable. The external dataset sasliba.Controls will be constructed to contain CCHS data from all of Canada. It is decided to select a weighted total of 500 controls for each case (frequency matched) by province. So if there are 30 cases in a province, controls will be randomly selected from the CCHS until their weighted total is 30*500=15000 (about 50 control records each with an average sample weight of 300). The appropriate numbers of controls are randomly sampled from the CCHS, weighting the probability of selection according to the CCHS sample weight.

The variable Prov will now appear on the controls dataset, coded according to the HPOI coding of Prov. The HPOIDD_Episode call must also specify that we want to retain the province code, and in the SAS code file we create Prov equaling _HPOIDD_Prov and remove the lines that subsetted only Ontario. Then the HPOIDD_Episode call, SAS code file and conditional logistic regression on these matched data can proceed as follows. (Code that is different than the previous call is in bold.)

HPOIDD_Episode call:

```
%HPOIDD_Episode(
     loclib.HPOIDD_BigData92to03,
     SASLIB1.CaseRheumLung,
     d:\bin\CaseRheumLung.txt,
     d:\bin\CaseRheumLung_SAScode.txt,
     RLng,
     1,
     FEDV_ADM,
     2000.01.01-2001.12.31,
     9999 weeks,
     AllVs,
     SEP to SEP NEGATIVES_TO_0,
     _NoSWV,
     _NoSWV,
     Count|ADMDATE|_HPOIDD_Prov*Person*POI_Dup|TotalTime,
     _FirstV_EpiEDVIR_RLng_BTHDATE _FirstV_EpiEDVIR_RLng_Sex
     _FirstV_EpiEDVIR_RLng_Prov
     );
```

Contents of d:\bin\CaseRheumLung_SAScode.txt:

```
*** Use all of Canada;
*** Shorter name Prov;
Prov=_HPOIDD_Prov;

*** The RLng EDV variable;
RLng9CM=0;
RLng10=0;
* Only a tabulating, or first diagnosis of RLng is counted;
do i=1 to 1; *** Not i=1 to _HPOIDD_DIAGNOSIS_ALEN;
 RLng9CM=RLng9CM+(substr(scan(DIAG_CM_CODE{i},1,' '),1,6) eq "714.81");
 RLng10=RLng10+(upcase(substr(scan(DIAG_ICD10_CODE{i},1,' '),1,5)) in
("M05.1"));
end;
RLng=(RLng9CM+RLng10 gt 0);
```

SAS code to perform conditional logistic regression:

```
data cases (keep=fwgt Caseness Age1994 Sex Prov) suspect;
 set saslib1.CaseRheumLung;
 /*** With a washout of 9999 weeks no-one should have more than one
      EDV visit ***/
 if _Count_RLng gt 1 then output suspect;

 /*** Keep only cases, and set weight to 1 since each case
```

```
      represents only himself or herself ***/
 if _Count_RLng eq 1;
 Caseness=1;
 fwgt=1;
 /*** Exposure variables ***/
 Sex=_FirstV_EpiEDVIR_RLng_Sex;
 Prov=_FirstV_EpiEDVIR_RLng_Prov;
 if Sex eq 0 then Sex=.; * Set unknown sex to missing;
 Age1994=(mdy(1,1,1994)- _FirstV_EpiEDVIR_RLng_BTHDATE)/365.25;
 output cases;
 run;

data usedat; set cases sasliba.Controls; run;

proc logistic descending data=usedat;
 title1 "Example of Conditional logistic regression ";
 title2 "On matched case-control data";
 strata Prov;
 class Sex / param=ref;
 model Caseness=Sex Age1994;
 weight fwgt;
 run;

proc freq data=usedat;
 title1 " Stratified Mantel-Haenszel odds ratio for sex versus rheumatic
lung";
 title2 "On matched case-control data";
 title3 "(Result may differ from above logistic analysis since;
 title4 "effect of sex is not adjusted for age group)";
 tables Prov*Sex*Caseness / cmh;
 weight fwgt;
 run;
```

The class statement in the Proc Logistic call tells SAS that Sex is a categorical variable and to use reference cell coding. The model coefficients from this model estimate the effect on the log odds of an RLng episode of each exposure variable controlling for the other, and controlling for the matching variable province. The stratified Mantel-Haenszel odds ratio estimated in the Proc Freq call can be used as a rough comparison. However, it should be noted that the estimated effect of sex in this call has not been adjusted for age group, and therefore some difference may be expected.

Example 3 – Person-level case-control Count data matched by a propensity score (using CCHS for controls)

Suppose the analyst instead wished to match cases and controls on the basis of a propensity score. This is generally the propensity to possess the attribute or exposure under study. In order to do this, the analyst would create an auxiliary dataset with the person identifier variables _HPOIDD_Prov, Person and POI_Dup, and the calculated propensity score. That dataset would be merged by _HPOIDD_Prov*Person*POI_Dup with the output HPOIDD_Episode dataset. The pool of controls (e.g., CCHS) would also have this score calculated per subject. Then similar to how controls were selected according to province when matching was done by province, controls would be sampled by propensity score such that a weighted total of 500 controls was (frequency) matched to each case by propensity score bins. So if there were 30 cases in a given

bin (interval) of propensity score, controls would be randomly selected from the subset of the CCHS in that bin until their weighted total was 30*500=15000 (about 50 control records each with an average sample weight of 300). The appropriate numbers of controls would be randomly sampled from the CCHS, weighting the probability of selection according to the CCHS sample weight. The binned (categorical) propensity score would be on the case and the control datasets. The two datasets would be appended, and conditional logistic regression on these matched data could proceed as before, but this time conditioning on propensity score bin rather than province.

6.7 Event-time models

Event-time models are concerned with analyzing the time to occurrence of an event. These data are also commonly called lifetime or failure time data because in many studies the event being modeled is the death of the subject or failure of the product. Methods of analysis include non-parametric analyses such as Kaplan-Meier (perhaps stratified and analyzed in part with the log rank test) or life table, semi-parametric methods such as the Cox proportional hazards model, and fully parametric regression models such as exponential or Weibull regression[11].

Event-time data typically consist of a mixture of observed event times and censoring times. Censoring can be in the form of left censoring in which it is only known that the event occurred prior to the recorded time, or right censoring in which it is only known that the event occurred after the recorded time. Interval censoring is a situation in which it is known only that the event occurred between two times. Even though one does not know the precise time of an event in censored data, those records still make valuable contributions to the analysis. All mainstream event-time modeling methods make use of censored records.

With HPOIDD, event-time models can be fit under various data designs.

- First (and perhaps least ideal), event-time models could be run on unlinked person-level HPOI data. This would assume that the relevant population consists of those people hospitalized for some reason (and also discharged) in the available years of data as the only subjects in the analysis would have to be those found in HPOI data years.
- Event-time models could also be fit to person-level HPOI data linked to an external data source such as a national population health survey, retaining in the analysis only those persons in the national survey sample, and subjects in the survey sample not found in HPOI or not experiencing the episode would represent censored times. Such a sample would be representative of Canada. However, the caveats around event time modeling explained in section 6.1, subsection *Caveats of HPOI data* still apply.
- Another alternative is to fit event-time models to unlinked HPOI data that is either hospital- or health region-level data, or perhaps provincial. Assuming that all hospitals or health regions or provinces are in the HPOI data years, this sample (it is a sample at least in time—in experimental units it is a census) would be representative of Canada. However, the caveats around event time modeling explained in section 6.1, subsection *Caveats of HPOI data* still apply.
- Perhaps the most powerful HPOIDD_Episode data design available for event time models is the EpisodeArray data design, where the output data consist of one or more episodes of multiple definitions all stored in a single array with one person per record. This is useful for analyses involving questions amongst several different episode types simultaneously, such as (generically) length of time to an episode of type A from the end

of an episode of type B. One example could be time to death from an episode of AMI (acute myocardial infarction), or time to pacemaker implantation from AMI and then time to death following pacemaker. Another example could be time to death from an episode of AMI, or competing risks between time to pacemaker implantation from AMI and time to death from AMI.

Example 1 – Life table analysis, stratified Kaplan-Meier with log rank test, Cox proportional hazards model, and parametric regressions with exponential and Weibull distributions, on person-level EventTime data linked to CCHS

Suppose the analyst wishes to compare the event-time distribution to "implantation, removal or replacement of cardiac pacemaker" starting measurement in calendar year 2000 (no assumption is made about previous events), between males and females in Ontario. This procedure has various codes according to the Canadian Classification of Procedures (CCP) (HPOIDD_BigData variables INTERVENTION_CCP_CODE{i}), Canadian Classification of Interventions (CCI) (HPOIDD_BigData variables INTERVENTION_CCI_CODE{i}) and ICD-9-CM (HPOIDD_BigData variables INTERVENTION_CM_CODE{i}). Suppose that in the models that allow covariates, the analyst also wants to adjust for comorbidity, age and a baseline variable representing the propensity of a person to be admitted into acute hospitals using all visits before January 1, 2000. In this example we will show code for life table analysis, stratified Kaplan-Meier with log rank test, Cox proportional hazards model, and parametric regressions with exponential and Weibull distributions. We link the data to the 2000/2001 CCHS in order to include as right-censored observations persons who do not appear in the available HPOI data, and this also means the analysis dataset will be representative of the Ontario population. We restrict our analysis dataset to those who appear in the Ontario portion of the CCHS.

(The analyst understands that the only admissions counted in the data are those that are subsequently discharged within the available years of data.)

Warning:
The ICD-9, ICD-9-CM, ICD-10, CCP and/or CCI codes presented in this example are for structural and syntactical illustration only and may not be accurate codes for the diseases and/or procedures in this example.

The call to the HPOIDD_Episode macro is set up as follows.

- The input dataset is specified via INDAT as loclib.HPOIDD_BigData92to03 (created above in section *6.2 Example of preparing the HPOIDD_BigData dataset*).
- The output SAS dataset is specified in OUTDAT to be saved in SASLIB1.EventTimePace.
- The output text readme file for the output dataset is specified in OUTTEXT to be saved in d:\bin\EventTimePace.txt.
- The SAS code defining the subgroup of interest and the episodes is defined in SASCODEPATH to be located in d:\bin\EventTimePace_SAScode.txt. In addition to defining the 0-1 EDV variables, the SAS code also uses only the Ontario subset of the HPOIDD_BigData dataset.

- There are two EDV variables required. The event type is named Pace, and the dummy episode type we call Baseline. Both are named in the EDVNAME argument and these 0-1 EDV variables are defined in the SAS code, Pace defined under CCP, CCI and CM intervention codes, and Base defined according to SEPDATE being on or before December 31, 1999.
- At least 1 episode-defining visit (EDV) with variable value of Pace=1 is required to count the Pace episode, specified via EDVMIN. At least 1 episode-defining visit (EDV) with variable value of Base=1 is required to count the Base episode, specified via EDVMIN.
- Pace episode date is considered to be the first EDV's admission date in a Pace episode, while Base episode date is considered to be the last visit's (EDV or not) separation date in a Base episode, both specified via EPIOCCUR.
- Only Pace episodes (ADMDATE of first EDV) occurring from January 1, 2000 forward are counted, specified via DATERANGE. Only Base episodes (SEPDATE of last visit) occurring up to and including December 31, 1999 are counted. Both ranges are specified via DATERANGE.
- For Pace, we set WASHTIME=1 days, WASHTYPE=AllVs and WASHCOMP=SEP TO ADM, to allow transfers for any reason directly between institutions (plus or minus a day for error) to represent continuations of an episode of care. For Base, we set WASHTIME=9999 weeks, WASHTYPE=EDVS (this is critical to stop the "baseline" measurement episode on the last visit ending before the year 2000) and WASHCOMP=SEP TO ADM, to allow all visits before 2000 to contribute to the baseline variables.
- There are no special washout visit (SWV) variables specified. This is indicated via SWVNAME and SWVLOGIC set to _NOSWV.
- DATADESIGN specifies the data design. The EventTime data design is specified. For the purposes of calculating in-range summary analysis variables specified on AVARLIST, and for determining whether a visit is in-range or not, a visit is deemed to occur on ADMDATE. The experimental unit is person defined by _HPOIDD_Prov*Person*POI_Dup. The time unit of the event-time analysis is Days.
- There is a user-defined 0-1 comorbidity variable defined in the SAS code in order to control for potential confounding between acute care facility and the comorbid conditions amongst diagnoses in the categories "DISEASES OF ORAL CAVITY, SALIVARY GLANDS, AND JAWS" (ICD-9/ICD-9-CM codes 520-529, or ICD-10 codes K00-K14) and "DISEASES OF ESOPHAGUS, STOMACH, AND DUODENUM" (ICD-9/ICD-9-CM codes 530-538, or ICD-10 codes K20-K31). Another user-defined variable in this example is HTypeAcute set to 1 if Hospital_Type indicates acute hospital type, else 0.
- An auxiliary dataset called sasliba.AuxInfo derived from the Ontario portion of the CCHS is available with a survey weight variable FWGT and a set of 1000 bootstrap weights BSW1-BSW1000. Linking variables are also on the dataset: _HPOIDD_Prov, Person and POI_Dup.
- AVARLIST specifies the analysis summary variables to include on the output dataset, that will describe the properties of variables during event (first in-range) episodes under the EventTime data design. _Mean_EpiEDV_Base_HTypeAcute contains the baseline proportion of acute hospital type across all visits by that person prior to the year 2000. _Mean_EpiVis_Base_UDComor contains the mean of the 0-1 comorbidity variable amongst all baseline visits for that person prior to the year 2000. Since there will not be

an episode of pacemaker for many persons, we will also use the Base episode to gather age and sex information (rather than a summary analysis variable on event pacemaker episode). Thus, we have _LastV_EpiEDV_Base_Sex and _LastV_EpiEDV_Base_BTHDATE to contain last (pre-year 2000) measured sex and birth date.

```
%HPOIDD_Episode(
     loclib.HPOIDD_BigData92to03,
     SASLIB1.EventTimePace,
     d:\bin\EventTimePace.txt,
     d:\bin\EventTimePace_SAScode.txt,
     Base|Pace,
     1|1,
     LV_SEP|FEDV_ADM,
     1900.01.01-1999.12.31|2000.01.01-2075.01.01,
     9999 weeks|1 days,
     EDVs|AllVs,
     SEP to ADM|SEP to ADM,
     _NoSWV,
     _NoSWV,
     EventTime|ADMDATE|_HPOIDD_Prov*Person*POI_Dup|Days,
     _Table_EpiEDV_Base_HTypeAcute _LastV_EpiEDV_Base_Sex
     _LastV_EpiEDV_Base_BTHDATE _Mean_EpiVis_Base_UDComor
     );
```

The contents of d:\bin\EventTimePace_SAScode.txt are:

```
*** Use only the Ontario subset of the HPOIDD_BigData dataset;
if _HPOIDD_Prov eq 35;

*** The dummy baseline visit indicator;
if SEPDATE le mdy(12,31,1999) then Base=1;
else Base=0;

*** Pacemaker procedure;
Pace=0;
do i=1 to _HPOIDD_INTERVENTION_ALEN;
if scan(INTERVENTION_CCP_CODE{i},1,' ') in
        ("49.7" "49.81" "49.82" "49.83" "49.84" "49.88") or
   scan(INTERVENTION_CM_CODE{i},1,' ') in
        ("37.7" "37.97" "37.75" "37.76" "37.85" "37.86" "37.87" "37.89" "37.99") or
   scan(INTERVENTION_CCI_CODE{i},1,' ') in
        ("1.HB.53" "1.HD.53" "1.HZ.53" "1.HB.54" "1.HD.54" "I.HZ.54" "1.HZ.55")
  then Pace=1;
end;

*** 0-1 indicator for Acute Hospital_Type;
HTypeAcute=(Hospital_Type eq "1");

*** User-defined 0-1 comorbidity variable UDComor;
UDComor9=0;
UDComor9CM=0;
UDComor10=0;
do i=1 to _HPOIDD_DIAGNOSIS_ALEN;
```

```
 UDComor9=UDComor9+(substr(scan(DIAG_ICD9_CODE{i},1,' '),1,3) in
          ("520" "521" "522" "523" "524" "525" "526" "527" "528" "529" "530"
           "531" "532" "533" "534" "535" "536" "537" "538"));
 UDComor9CM=UDComor9CM+(substr(scan(DIAG_ICD9_CODE{i},1,' '),1,3) in
           ("520" "521" "522" "523" "524" "525" "526" "527" "528" "529" "530"
            "531" "532" "533" "534" "535" "536" "537" "538"));
 UDComor10=UDComor10+(upcase(substr(scan(DIAG_ICD10_CODE{i},1,' '),1,3)) in
            ("K00" "K01" "K02" "K03" "K04" "K05" "K06" "K07" "K08"
             "K09" "K10" "K11" "K12" "K13" "K14" "K20" "K21" "K22"
             "K23" "K24" "K25" "K26" "K27" "K28" "K29" "K30" "K31"));
end;
UDComor=(UDComor9+UDComor9CM+UDComor10 gt 0);
```

The output dataset will contain the following variables.

- The experimental unit identifiers, in this example _HPOIDD_Prov, Person and POI_Dup.
- The variables _FstDateRsk_Base, _EventDate_Base, _EventDays_Base, _Censored_Base, _FstDateRsk_Pace, _EventDate_Pace, _EventDays_Pace and _Censored_Pace are produced automatically when the data design type EventTime is specified.
- Since the experimental unit is person defined by _HPOIDD_Prov*Person*POI_Dup, there will also be _NumEDV_Base, _NumEDV_Pace, _NumAllV_Base, _NumAllV_Pace, _DistinctDays_Base, _DistinctDays_Pace, _OvercountDays_Base and _OvercountDays_Pace.
- Whatever special analysis summary variables are requested, in this case _Table_EpiEDV_Base_HTypeAcute, _Mean_EpiVis_Base_UDComor, _LastV_EpiEDV_Base_Sex and _LastV_EpiEDV_Base_BTHDATE.

The following SAS code shows how to link these data, and analyze them via life table analysis, stratified Kaplan-Meier with log rank test, Cox proportional hazards model, and parametric regressions with exponential and Weibull distributions.

```
/*** saslib1.EventTimePace should already be sorted by
     _HPOIDD_Prov*Person*POI_Dup ***/
proc sort data=sasliba.AuxInfo; by _HPOIDD_Prov Person POI_Dup; run;
data usedat;
 merge saslib1.EventTimePace sasliba.AuxInfo;
 by _HPOIDD_Prov Person POI_Dup;
 run;
data usedat
  (keep=fwgt bsw1-bsw1000 _HPOIDD_Prov Person POI_Dup
        _EventDays_Pace _Censored_Pace
        Sex Age2000 _Mean_EpiVis_Base_UDComor _Mean_EpiVis_Base_HTypeAcute);
 set usedat;
 /*** Only keep records from the Ontario portion of the CCHS ***/
 if fwgt ne .;
 /*** Stratification variable and covariates ***/
 if _LastV_EpiEDV_Base_Sex in (1 2) then Sex=_LastV_EpiEDV_Base_Sex;
 else sex=.;
 Age2000=(mdy(1,1,2000)-_LastV_EpiEDV_Base_BTHDATE)/365.25;
 run;

/* WARNING
```

```
   We interpret confidence intervals with caution due to the weight
   being treated as a frequency except where the norm option is
   available, and in that case caution is warranted since CCHS and NPHS
   have complex survey designs. Bootstrapping may be done on
   many procedures to obtain proper CIs, and/or in the case
   of weight statements without norm options, scaling the weight
   to sum to the sample size can improve the SE estimates though not
   account for complex designs. Consult your bootstrap software. */

/* The proc lifetest analyses do not control for
   Age2000 _Mean_EpiVis_Base_UDComor _Mean_EpiVis_Base_HTypeAcute */
   To control for covariates, one could create an additional
   categorical variable on which to stratify.

proc lifetest data=usedat method=lt;
 title1 "Example of stratified life table analysis on";
 title2 "person-level event-time data linked to CCHS";
 strata Sex;
 time _EventDays_Pace*_Censored_Pace(1);
 freq fwgt;
 run;

proc lifetest data=usedat method=km;
 title1 "Example of stratified Kaplan-Meier analysis with log rank test on";
 title2 "person-level event-time data linked to CCHS";
 strata Sex / test=logrank;
 time _EventDays_Pace*_Censored_Pace(1);
 freq fwgt;
 run;

/* The remaining analyses do control for
   Age2000 _Mean_EpiVis_Base_UDComor _Mean_EpiVis_Base_HTypeAcute */

proc phreg data=usedat;
 title1 "Example of Cox proportional hazards regression on";
 title2 "person-level event-time data linked to CCHS";
 model _EventDays_Pace*_Censored_Pace(1)=
          Sex Age2000 _Mean_EpiVis_Base_UDComor _Mean_EpiVis_Base_HTypeAcute;
 weight fwgt / norm;
 run;

proc lifereg data=usedat;
 title1 "Example of Exponential regression on";
 title2 "person-level event-time data linked to CCHS";
 model _EventDays_Pace*_Censored_Pace(1)=
          Sex Age2000 _Mean_EpiVis_Base_UDComor _Mean_EpiVis_Base_HTypeAcute
          / dist=exponential;
 weight fwgt;
 run;

proc lifereg data=usedat;
 title1 "Example of Weibull regression on";
 title2 "person-level event-time data linked to CCHS";
 model _EventDays_Pace*_Censored_Pace(1)=
          Sex Age2000 _Mean_EpiVis_Base_UDComor _Mean_EpiVis_Base_HTypeAcute
          / dist=Weibull;
 weight fwgt;
```

```
 run;
```

Example 2 – Cox proportional hazards model with time varying covariates on EventTime data in person-level data linked to NPHS

Suppose the analyst again wishes to analyze event-time distribution to "implantation, removal or replacement of cardiac pacemaker", but this time with explanatory variables age, sex and use of the Coxib class of drugs. While age and sex are available in HPOI data, detailed drug information is not, nor is it on the CCHS. It is however available in the (1994 to 2004) longitudinal NPHS data. Therefore linking the data to the NPHS would be advantageous. As before, this means the analysis dataset will be representative of the Canadian population, and it will include as right-censored observations persons who do not appear in the available HPOI data. Since the NPHS is smaller than the CCHS and pacemaker procedures are not that common, we do not take the Ontario subset but use all of Canada. As before, only those who are in the NPHS sample can be used. In this example we show code for the Cox proportional hazards model with time-varying covariates.

(The analyst understands that the only admissions counted in the data are those that are subsequently discharged within the available years of data.)

Warning:
The ICD-9, ICD-9-CM, ICD-10, CCP and/or CCI codes presented in this example are for structural and syntactical illustration only and may not be accurate codes for the diseases and/or procedures in this example.

The call to the HPOIDD_Episode macro is set up the same as before. The only difference is the available auxiliary dataset.

- An auxiliary dataset called sasliba.AuxInfo derived from the NPHS is available with a survey weight variable FWGT and a set of 1000 bootstrap weights BSW1-BSW1000. Linking variables are also on the dataset: _HPOIDD_Prov, Person and POI_Dup. In addition, a derived 0-1 indicator for use of the Coxib class of drugs is included for each cycle: bCoxib2000, bCoxib2002 and bCoxib2004.

The contents of d:\bin\EventTimePace_SAScode.txt are almost the same as in the previous example, except that we omit the line at the top of the program that subsets the Ontario data.

```
*** Use no longer use only the Ontario subset of the HPOIDD_BigData dataset;
/*** Commented out or deleted: if _HPOIDD_Prov eq 35; ***/
```

The following SAS code shows how to link these data, and analyze them via the Cox proportional hazards model with time-varying covariates.

```
/*** saslib1.EventTimePace should already be sorted by
     _HPOIDD_Prov*Person*POI_Dup ***/
proc sort data=sasliba.AuxInfo; by _HPOIDD_Prov Person POI_Dup; run;
data usedat;
 merge saslib1.EventTimePace sasliba.AuxInfo;
 by _HPOIDD_Prov Person POI_Dup;
 run;
```

```
data usedat
  (keep=fwgt bsw1-bsw1000 _HPOIDD_Prov Person POI_Dup
        _EventDays_Pace _Censored_Pace
        Sex Age2000 _Mean_EpiVis_Base_UDComor _Mean_EpiVis_Base_HTypeAcute
        _LastV_EpiEDV_Base_BTHDATE
        bCoxib2000 bCoxib2002 bCoxib2004);
 set usedat;
 /*** Only keep records from the NPHS ***/
 if fwgt ne .;
 /*** Stratification variable and covariates ***/
 if _LastV_EpiEDV_Base_Sex in (1 2) then Sex=_LastV_EpiEDV_Base_Sex;
 else sex=.;
 run;

/* WARNING
   We interpret confidence intervals with caution due to the weight
   being treated as a frequency except where the norm option is
   available, and in that case caution is warranted since CCHS and NPHS
   have complex survey designs. Bootstrapping may be done on
   many procedures to obtain proper CIs, and/or in the case
   of weight statements without norm options, scaling the weight
   to sum to the sample size can improve the SE estimates though not
   account for complex designs. Consult your bootstrap software. */

proc tphreg data=usedat;
 title1 "Example of Cox proportional hazards regression";
 title2 "with time-varying covariates";
 title2 "on person-level event-time data linked to NPHS";
 class Sex;
 model _EventDays_Pace*_Censored_Pace(1)=
          Age bCoxib
          Sex _Mean_EpiVis_Base_UDComor _Mean_EpiVis_Base_HTypeAcute;
 Age=(mdy(1,1,2000)+_EventDays_Pace-_LastV_EpiEDV_Base_BTHDATE)/365.25;
 if year(_EventDays_Pace+mdy(1,1,2000)) lt 2002 then bCoxib=bCoxib2000;
 else if year(_EventDays_Pace+mdy(1,1,2000)) lt 2004 then bCoxib=bCoxib2002;
 else bCoxib=bCoxib2004;
 weight fwgt / norm;
 run;
```

Example 3 – Event time modeling from first hospital admission to the next (uses the EpisodeLevel data design)

Suppose the analyst wishes to analyze the event-time distribution from the first "implantation, removal or replacement of cardiac pacemaker" in calendar year 1994 of the data or later to the next such operation (measuring alive failure rate, or rate of failed pacemakers with patients who make it back into hospital alive for replacement), on person-level data, with explanatory variables age and sex, and the analysis done by province. It is assumed that persons not returning for a second pacemaker operation in the data are right censored on March 31, 2004. The limitations of this assumption will be written up along with the findings. This problem could be tackled using the EpisodeArray data design, but for illustrative purposes we show a solution using the EpisodeLevel data design. An example of an analysis using the EpisodeArray data design is presented next.

(The analyst understands that the only admissions counted in the data are those that are subsequently discharged within the available years of data.)

Warning:
The ICD-9, ICD-9-CM, ICD-10, CCP and/or CCI codes presented in this example are for structural and syntactical illustration only and may not be accurate codes for the diseases and/or procedures in this example.

The call to the HPOIDD_Episode macro is set up as follows.

- The input dataset is specified via INDAT as loclib.HPOIDD_BigData92to03 (created above in section *6.2 Example of preparing the HPOIDD_BigData dataset*).
- The output SAS dataset is specified in OUTDAT to be saved in SASLIB1.EpisodesPace.
- The output text readme file for the output dataset is specified in OUTTEXT to be saved in d:\bin\EpisodesPace.txt.
- The SAS code defining the subgroup of interest and the episodes is defined in SASCODEPATH to be located in d:\bin\EpisodesPace_SAScode.txt.
- The single episode type of interest is named Pace in the EDVNAME argument and this 0-1 EDV variable is defined in the SAS code under CCP, CCI and CM intervention codes.
- At least 1 episode-defining visit (EDV) with variable value of Visit=1 is required to count the episode, specified via EDVMIN.
- Episode date is considered to be the first EDV's admission date in the episode, specified via EPIOCCUR.
- Only episodes (first EDV admissions) occurring between January 1, 1994 and March 31, 2004 are counted, specified via DATERANGE.
- We set WASHTIME=0 days, WASHTYPE=EDVS and WASCOMP=SEP TO ADM NEGATIVES_TO_0 to ensure that only the pacemaker procedure visits are counted and no two visits are joined into one episode.
- There are no special washout visit (SWV) variables specified. This is indicated via SWVNAME and SWVLOGIC set to _NOSWV.
- DATADESIGN specifies the data design. The EpisodeLevel data design is specified. The experimental unit under the EpisodeLevel data design is episode, which automatically occurs within person defined by _HPOIDD_Prov*Person*POI_Dup. So there will be one or more records per person in the output data.
- There are no user-defined variables in this example.
- AVARLIST specifies the analysis summary variables to include on the output dataset. _LastV_EpiVis_Pace_BTHDATE and _LastV_EpiVis_Pace_Sex will contain (for each person in the output dataset) the birth date and sex assessed at separation from the single EDV visit for that episode. We also request _FirstV_EpiVis_Pace_SEPDATE which is the separation date of the only EDV visit in the episode. As a data validity check we also request _FirstV_EpiVis_Pace_ADMDATE. Since there is only one visit (and it is an EDV) per episode, and we made first EDV ADMDATE the episode occurrence date, then the variable _EpiDate should equal _FirstV_EpiVis_Pace_ADMDATE.

```
%HPOIDD_Episode(
     loclib.HPOIDD_BigData92to03,
     SASLIB1.EpisodesPace,
```

```
     d:\bin\EpisodesPace.txt,
     d:\bin\EpisodesPace_SAScode.txt,
     Pace,
     1,
     FEDV_ADM,
     1994.01.01-2004.03.31,
     0 days,
     EDVS,
     SEP to ADM NEGATIVES_TO_0,
     _NoSWV,
     _NoSWV,
     EpisodeLevel,
     _LastV_EpiVis_Pace_BTHDATE _LastV_EpiVis_Pace_Sex
     _FirstV_EpiVis_Pace_SEPDATE _FirstV_EpiVis_Pace_ADMDATE
     );
```

The contents of d:\bin\EpisodeLevelStay_SAScode.txt are:

```
*** Pacemaker procedure;
Pace=0;
do i=1 to _HPOIDD_INTERVENTION_ALEN;
if scan(INTERVENTION_CCP_CODE{i},1,' ') in
        ("49.7" "49.81" "49.82" "49.83" "49.84" "49.88") or
   scan(INTERVENTION_CM_CODE{i},1,' ') in
        ("37.7" "37.97" "37.75" "37.76" "37.85" "37.86" "37.87" "37.89" "37.99") or
   scan(INTERVENTION_CCI_CODE{i},1,' ') in
        ("1.HB.53" "1.HD.53" "1.HZ.53" "1.HB.54" "1.HD.54" "I.HZ.54" "1.HZ.55")
  then Pace=1;
end;
```

The output dataset will contain the following variables.

- In EpisodeLevel data there is one record per person-episode. To identify person there are the person identifiers _HPOIDD_Prov, Person and POI_Dup. To identify episode there are the variables _EpisodeType and _EpiDate.
- The variables _NumALLV, _NumEDV, _DistinctDays and _OvercountDays are also produced automatically when the data design type EpisodeLevel is specified.
- Whatever special analysis summary variables are requested, in this case _LastV_EpiVis_Pace_BTHDATE, _LastV_EpiVis_Pace_Sex, _FirstV_EpiVis_Pace_SEPDATE and _FirstV_EpiVis_Pace_ADMDATE.

The following SAS code shows how to organize and analyze these data to address the questions, via the Cox proportional hazards model.

```
/*** SASLIB1.saslib1.EpisodesPace should already be sorted by
     _HPOIDD_Prov*Person*POI_Dup*_EpiDate ***/
data usedat
  (keep=_HPOIDD_Prov Person POI_Dup
        BaseAge Sex EventDays Censored)
     bad;
 set saslib1.EpisodesPace;
 by _HPOIDD_Prov Person POI_Dup _EpiDate;
 retain BaseAge Sex EventDays Censored Base_EpiDate bOut;
```

```
 if first._HPOIDD_Prov or first.Person or first.POI_Dup then do;
  if _LastV_EpiVis_Pace_Sex in (1 2) then Sex=_LastV_EpiVis_Pace_Sex;
  else Sex=.;
  BaseAge=(_FirstV_EpiVis_Pace_SEPDATE-_LastV_EpiVis_Pace_BTHDATE)/365.25;
  Base_EpiDate=_EpiDate;
  bOut=0;
  Censored=.;
  EventDays=.;
  *** If the first episode is also the last then right-censored;
  if last._HPOIDD_Prov or last.Person or last.POI_Dup then do;
   Censored=1;
   EventDays=mdy(3,31,2004)-Base_EpiDate;
   bOut=1;
   *** Data integrity check;
   if _FirstV_EpiVis_Pace_ADMDATE ne _EpiDate then output bad;
   else output usedat;
  end;
 end;

 *** Be careful! This code treats the next encountered pacemaker procedure;
 *** as the event. If there are more than one subsequent procedure,;
 *** they are not included in this particular analysis.;
 if ~(first._HPOIDD_Prov or first.Person or first.POI_Dup)
     and bOut eq 0 then do;
  Censored=0;
  EventDays=_EpiDate-Base_EpiDate;
  bOut=1;
  *** Data integrity check;
  if _FirstV_EpiVis_Pace_ADMDATE ne _EpiDate then output bad;
  else output usedat;
 end;
 run;

proc sort data=usedat; by _HPOIDD_Prov; run;
proc tphreg data=usedat;
 title1 "Example of Cox proportional hazards regression";
 title2 "on person-level pacemaker replacement operation";
 class Sex;
 model EventDays*Censored(1)=BaseAge sex;
 by _HPOIDD_Prov;
 run;
```

Example 4 – Competing risks event time modeling from the first episode of one type to the first of several competing episodes (uses the EpisodeArray data design)

Suppose the analyst wishes to perform a competing risks analysis of the event-time distribution starting from admission for acute myocardial infarction (AMI) not following within 1 year an admission from a visit with a diagnosis of "arterial embolism and thrombosis", to the first event between death and "implantation, removal or replacement of cardiac pacemaker". Note that this analyst wants to omit the AMI episode even if the visit for "arterial embolism and thrombosis" is the same visit as the AMI episode. This is not done automatically using SWV settings. The analyst must add the additional clause to their SAS code to preclude AMI visits from being AMI EDVs if they occur at the same time as the SWV. The time period of interest is from January 1,

1994. Explanatory variables in this example are age and sex. It is assumed that persons not experiencing a pacemaker operation and who do not die in the data are right censored on March 31, 2004. The limitations of this assumption will be written up along with the findings. For this problem we utilize the EpisodeArray data design.

(The analyst understands that the only admissions counted in the data are those that are subsequently discharged within the available years of data.)

Warning:
The ICD-9, ICD-9-CM, ICD-10, CCP and/or CCI codes presented in this example are for structural and syntactical illustration only and may not be accurate codes for the diseases and/or procedures in this example.

The call to the HPOIDD_Episode macro is set up as follows.

- The input dataset is specified via INDAT as loclib.HPOIDD_BigData92to03 (created above in section *6.2 Example of preparing the HPOIDD_BigData dataset*).
- The output SAS dataset is specified in OUTDAT to be saved in SASLIB1.EpisodeArrayMixed.
- The output text readme file for the output dataset is specified in OUTTEXT to be saved in d:\bin\EpisodeArrayMixed.txt.
- The SAS code defining the subgroup of interest and the episodes is defined in SASCODEPATH to be located in d:\bin\EpisodeArrayMixed_SAScode.txt.
- The three episode types of interest are named AMIWOAT (this stands for AMI **without** "arterial embolism and thrombosis" and reminds the analyst of the additional condition they are placing in the SAS code to this effect), Pace and Death in the EDVNAME argument and these 0-1 EDV variables are defined in the SAS code. AMIWOAT is defined under ICD-9, ICD-9-CM and ICD-10 specifications. Pace is defined under CCP, CCI and CM intervention codes. Death is defined by DISCHARGE_DISP_POI eq "07" or "7".
- At least 1 episode-defining visit (EDV) of each episode type is required to count the episode, specified via EDVMIN.
- The episode date of AMIWOAT is considered to be the last visit's (EDV or not) separation date since the event time model will begin then. The episode date of Pace is considered to be the EDV's admission date since admission for the procedure is the pacemaker event. The episode date for Death is considered to be the EDV's separation date since that is the best estimate of when the patient died and hence of the death event. These are all specified via EPIOCCUR.
- Only AMIWOAT (last visit's (EDV or not) separation), Pace (first EDV admission) and Death (first EDV separation) episodes occurring between January 1, 1994 and March 31, 2004 are counted, specified via DATERANGE.
- For AMIWOAT episodes, we set WASHTIME=1 weeks, WASHTYPE=ALLVS and WASCOMP=SEP TO ADM to allow transfers between institutions to be counted as continuations of the episode as long as the readmission occurs within 1 week of the previous separation. For both Pace and Death episodes, we set WASHTIME=0 days, WASHTYPE=EDVS and WASCOMP=SEP TO ADM NEGATIVES_TO_0 to ensure

that only the EDV visits are counted in an episode and no two visits are joined into one episode.

- Special washout visit (SWV) settings specify that potential AMIWOAT visits with ADMDATE between -5 days after (i.e., 5 days before) and 52 weeks after the SEPDATE of a visit with a diagnosis code "arterial embolism and thrombosis" are precluded from being AMIWOAT EDVs. The SWV name AET is specified via SWVNAME, and this 0-1 SWV variable is defined in the SAS code under ICD-9, ICD-9-CM and ICD-10 specifications. The SWV logic is specified via SWVLOGIC.
- DATADESIGN specifies the data design. The EpisodeArray data design is specified. The experimental unit under the EpisodeArray data design is person defined by _HPOIDD_Prov*Person*POI_Dup. So there will be one record per person in the output data, as long as a person has at least one episode of either type.
- AVARLIST specifies the analysis summary variables to include on the output dataset. _FirstV_EpiVis_SEX and _FirstV_EpiVis_BTHDATE produce (where &maxepisodes is the maximum number of episodes for a single person of all episode kinds combined) _FirstV_EpiVis_SEX1-_FirstV_EpiVis_SEX&maxepisodes and _FirstV_EpiVis_BTHDATE1-_FirstV_EpiVis_BTHDATE&maxepisodes, containing sex and birth date from the first visit in each episode.

```
%HPOIDD_Episode(
     loclib.HPOIDD_BigData92to03,
     SASLIB1.EpisodeArrayMixed,
     d:\bin\EpisodeArrayMixed.txt,
     d:\bin\EpisodeArrayMixed_SAScode.txt,
     AMIWOAT|Pace|Death,
     1|1|1,
     LV_SEP|FEDV_ADM|FEDV_SEP,
     1994.01.01-2004.03.31|1994.01.01-2004.03.31|1994.01.01-2004.03.31,
     1 weeks|0 days|0 days,
     ALLVS|EDVS|EDVS,
     SEP TO ADM|SEP to ADM NEGATIVES_TO_0|SEP to ADM NEGATIVES_TO_0,
     AET,
     AET precludes AMIWOAT EDVADM-SWVADM from -5 days to 52 weeks,
     EpisodeArray,
     _FirstV_EpiVis_SEX _FirstV_EpiVis_BTHDATE
     );
```

The contents of d:\bin\EpisodeArrayMixed_SAScode.txt are:

```
*** Special washout variable (SWV) arterial embolism and thrombosis;
AET9=0;
AET9CM=0;
AET10=0;
* Any AET diagnosis is counted, first or not;
do i=1 to _HPOIDD_DIAGNOSIS_ALEN;
 AET9=AET9+(substr(scan(DIAG_ICD9_CODE{i},1,' '),1,3) eq "444");
 AET9CM=AET9CM+(substr(scan(DIAG_CM_CODE{i},1,' '),1,3) eq "444");
 AET10=AET10+(upcase(substr(scan(DIAG_ICD10_CODE{i},1,' '),1,3))
              eq "I74");
end;
AET=(AET9+AET9CM+AET10 gt 0);
```

```
*** AMIWOAT is a visit for AMI excluding a same visit with;
*** arterial embolism and thrombosis;
AMIWOAT9=0;
AMIWOAT9CM=0;
AMIWOAT10=0;
* Any AMIWOAT diagnosis is counted, first or not;
do i=1 to _HPOIDD_DIAGNOSIS_ALEN;
 AMIWOAT9=AMIWOAT9+(substr(scan(DIAG_ICD9_CODE{i},1,' '),1,3) eq "410");
 AMIWOAT9CM=AMIWOAT9CM+(substr(scan(DIAG_CM_CODE{i},1,' '),1,3) eq "410");
 AMIWOAT10=AMIWOAT10+(upcase(substr(scan(DIAG_ICD10_CODE{i},1,' '),1,3))
                      eq "I21");
end;
if AET eq 0 then AMIWOAT=(AMIWOAT9+AMIWOAT9CM+AMIWOAT10 gt 0);
else AMIWOAT=0;

*** Death;
if DISCHARGE_DISP_POI eq "07" or "7" then Death=1;
else Death=0;

*** Pacemaker procedure;
Pace=0;
do i=1 to _HPOIDD_INTERVENTION_ALEN;
if scan(INTERVENTION_CCP_CODE{i},1,' ') in
        ("49.7" "49.81" "49.82" "49.83" "49.84" "49.88") or
   scan(INTERVENTION_CM_CODE{i},1,' ') in
        ("37.7" "37.97" "37.75" "37.76" "37.85" "37.86" "37.87" "37.89"
"37.99") or
   scan(INTERVENTION_CCI_CODE{i},1,' ') in
        ("1.HB.53" "1.HD.53" "1.HZ.53" "1.HB.54" "1.HD.54" "I.HZ.54"
"1.HZ.55")
  then Pace=1;
end;
```

The output dataset will contain the following variables.

- In EpisodeArray data there is one record per person. To identify person there are the person identifiers _HPOIDD_Prov, Person and POI_Dup.
- Where &maxepisodes is the maximum number of episodes for a person in the data, _GrandMaxEpisodes will equal &maxepisodes, while _NumEpisodes will equal the number of episodes for that person (>=1). The variables _NumALLV1-_NumALLV&maxepisodes, _NumEDV1-_NumEDV&maxepisodes, _DistinctDays1-_DistinctDays&maxepisodes and _OvercountDays1-_OvercountDays&maxepisodes are also produced automatically when the data design type EpisodeArray is specified.
- Whatever special analysis summary variables are requested, in this case _FirstV_EpiVis_SEX1-_FirstV_EpiVis_SEX&maxepisodes and _FirstV_EpiVis_BTHDATE1-_FirstV_EpiVis_BTHDATE&maxepisodes.

The following SAS code shows how to analyze these data to address the research questions, via the competing risks Cox proportional hazards model.

```
%let maxepisodes=NULL;
data _null_;
 set saslib1.EpisodeArrayMixed;
 if _n_ eq 1 then call symput("maxepisodes",scan(_GrandMaxEpisodes,1,' '));
```

```
 run;
%let maxepisodes=%eval(&maxepisodes);

/*** Recall the warning under the AVARLIST argument about
     variables not ending in integers if we intend to use
     array statements. In this case Sex and BTHDATE do not end
     in integers so we were okay to specify them on AVARLIST
     without renaming them.
 ***/

data usedat
  (keep=_HPOIDD_Prov Person POI_Dup BaseAge Sex
        EventYears CensoredD CensoredP)
      bad;
set saslib1.EpisodeArrayMixed;
 array _EpiDate{&maxepisodes};
 array _EpisodeType{&maxepisodes};
 array _FirstV_EpiVis_SEX{&maxepisodes};
 array _FirstV_EpiVis_BTHDATE{&maxepisodes};
 _FirstAMIWOATi=0;
 _FirstPacei=0;
 _FirstDeathi=0;
 do i=1 to _NumEpisodes;
  if _FirstAMIWOATi eq 0 and _EpisodeName="AMIWOAT" then _FirstAMIWOATi=i;
  if _FirstPACEi eq 0 and _EpisodeName="PACE" then _FirstPACEi=i;
  if _FirstDEATHi eq 0 and _EpisodeName="DEATH" then _FirstDEATHi=i;
 end;
 if _FirstAMIWOATi ne 0 then do;
  BaseAge=_EpiDate{_FirstAMIWOATi}-_FirstV_EpiVis_BTHDATE{_FirstAMIWOATi};
  Sex=_FirstV_EpiVis_Sex{_FirstAMIWOATi};
  * Neiher Pace nor Death occurred after first AMIWOAT;
  if _FirstPacei eq 0 and _FirstDeathi eq 0 then do;
   EventYears=(mdy(3,31,2004)-_EpiDate{_FirstAMIWOATi})/365.25;
   CensoredDeath=1;
   CensoredPace=1;
   output usedat;
  end;
  * Only Pace occurred after first AMIWOAT;
  if _FirstPacei ne 0 and _FirstDeathi eq 0 then do;
   EventYears=(_EpiDate{_FirstPacei}-_EpiDate{_FirstAMIWOATi})/365.25;
   CensoredDeath=1;
   CensoredPace=0;
   output usedat;
  end;
  * Only Death occurred after first AMIWOAT;
  if _FirstPacei eq 0 and _FirstDeathi ne 0 then do;
   EventYears=(_EpiDate{_FirstDeathi}-_EpiDate{_FirstAMIWOATi})/365.25;
   CensoredDeath=0;
   CensoredPace=1;
   output usedat;
  end;
  * Both Pace and Death occurred after first AMIWOAT;
  if _FirstPacei ne 0 and _FirstDeathi ne 0 then do;
   * ADMDATE of Pace must occur before SEPDATE of Death;
   * for the data to be sensible;
   if _EpiDate{_FirstDeathi} lt _EpiDate{_FirstPacei} then output bad;
   else do;
```

```
    EventYears=(_EpiDate{_FirstPacei}-_EpiDate{_FirstAMIWOATi})/365.25;
    CensoredDeath=1;
    CensoredPace=0;
    output usedat;
   end;
  end;
 end;
 run;

proc tphreg data=usedat;
 title1 "Example of competing risks Cox proportional hazards regression";
 title2 "Submodel analyzing time from AMI to new pacemaker";
 title3 "censored at death or March 31, 2003";
 class Sex;
 model EventYears*CensoredPace(1)=Age Sex;
 run;

proc tphreg data=usedat;
 title1 "Example of competing risks Cox proportional hazards regression";
 title2 "Submodel analyzing time from AMI to death ";
 title3 "censored at new pacemaker or March 31, 2003";
 class Sex;
 model EventYears*CensoredDeath(1)=Age Sex;
 run;
```

7. References

1. Johansen H. Health Studies Using Linked Administrative Hospital Data. Proceedings of Statistics Canada Symposium. 2005.
2. Person Oriented Information and Hospital Morbidity Data Dictionary. Health Statistics Division, Statistics Canada. Prepared April, 1999, Updated March 27, 2003. File name: Hospital POI Data Dictionary.doc
3. Combined HPOI & HMDB Data Dictionary Data years: Fiscal 2001 to Fiscal 2004. Health Statistics Division, Statistics Canada. File name: Data_Dictionary_CANxxxx&Abstract_v2004.doc
4. Hospital Morbidity Database (HMDB) From Fiscal 2001 HMDB Data Dictionary for: Diagnosis Table. Health Statistics Division, Statistics Canada. File name: Data_Dictionary_Diagnosis_v2004.doc
5. Hospital Morbidity Database (HMDB) From Fiscal 2001 HMDB Data Dictionary for: Intervention Table. File name: Data_Dictionary_Intervention_v2004.doc
6. Dobson AJ. An Introduction to Generalized Linear Models. Chapman & Hall. London, UK. 1990.
7. Hosmer DW, Lemeshow S. Applied Logistic Regression. John Wiley & Sons, Inc. USA. 1989.
8. Weisberg, S. Applied Linear Regression. John Wiley & Sons, Inc. Hoboken, NJ, USA. 2005.
9. Montgomery, DC. Design and Analysis of Experiments, 4th ed. John Wiley & Sons, Inc. USA. 1997. p. 146.
10. Rothman KJ, Greenland S. Modern Epidemiology, 2nd ed. Lippencott-Raven Publishers. USA. 1998.
11. Lawless JF. Statistical Models and Methods for Lifetime Data. John Wiley & Sons, Inc. Hoboken, NJ, USA. 2003.
12. Diggle PJ, Liang K, Zeger SL. Analysis of Longitudinal Data. Oxford University Press, Inc. New York, NY, USA. 2000.